kilobaud

MICROCOMPUTING

March 1981
USA $2.95 DM 9.80

T.M.

DESKTOP COMPUTERS

The H-89: Putting It All Together

Heath's Do-It-Yourself Computer

Which Business Computer Is Right for You? □ 6800 Secrets □ Electric Crayon Shows Its True Colors □ Cheap, Dumb Apple □ North Star Database Manager □ Twice-As-Quick Fix for the OSI □ PET Shorthand Compleat

SOLVE THE STORAGE SHORTAGE

with ten multi-user megabytes

CompuStar's 8-inch Winchester

Intertec's exclusive 255 User Controller

CompuStar Cable Assembly

If you could think of just one way to improve our phenomenally popular SuperBrain, what would it be? More disk storage? Well, we already thought about it. And for only a few thousand dollars for a whopping 10 megabytes of lightning-fast storage, it's nothing short of another major breakthrough! From the company that wrote the book on price/performance . . . Intertec.

Our New CompuStar™ 10 Megabyte Disk Storage System (called a DSS) features an 8 inch Winchester drive packaged in an attractive, compact desktop enclosure. Complete with disk, controller and power supply. Just plug it into the Z80 adaptor of your SuperBrain and turn it on.* It's so quiet, you'll hardly know it's there. But, you'll quickly be astounded with its awesome power and amazing speed.

The secret behind our CompuStar DSS is its unique controller/multiplexor. It allows many terminals to "share" the resources of a single disk. So, not only can you use the DSS with your SuperBrain, you can configure multiple user stations using our new series of CompuStar™ terminals, called Video Processing Units or VPU's™.

Four CompuStar VPU's are available. At prices starting at less than $2,500. Some models are designed to operate as stand-alone microcomputers, with internal disk storage. Just like your SuperBrain. Each model features its own 64K of RAM and can be "daisy-chained" into a powerful multi-user network. Just connect one VPU into the next. Using easy-to-install cable assemblies. Connect up to 255 users in a single system. One at a time. As you need them.

Whether you need an extra 10 megabytes for your SuperBrain or an enormous multi-user network, the CompuStar™ DSS solves your storage shortage problems. Sensibly. And economically. Plus, your investment is protected by a nationwide service network with outlets in most major U.S. cities. Providing efficient on-site or depot maintenance.

Get a demonstration of this extraordinary new system today. Call or write now for the name and address of your nearest CompuStar dealer.

INTERTEC DATA SYSTEMS. ↙3

2300 Broad River Rd. Columbia, S.C. 29210
(803) 798-9100 TWX: 810-666-2115

*Some models require hardware/software modification.

MICROCOMPUTING

APPLICATIONS

BUSINESS

CONSTRUCTION PROJECTS

GENERAL

GRAPHICS

HARDWARE MODIFICATIONS

Page 28.

Page 72.

Contents: March 1981

Volume V
No. 3

This month:

Our March issue features part one of a comprehensive article on the Heathkit All-In-One Computer, the H-89. Beginning on page 28, H-89 owner Martin Moore describes his experiences in constructing the machine and getting it up and running. If you've been contemplating the purchase of an All-In-One, you'll appreciate his commentary. Next month's follow-up article will take a look at H-89 software.

Building your own computer wasn't always so easy. Of course, back in the Dark Ages (1975), the *only* way to acquire a working micro was to do it yourself. Even then, you could expect to spend weeks or months debugging the system before it performed properly. One of the reasons Wayne Green started *Kilobaud Microcomputing* was to help the early pioneers share fixes and tips.

When Heath Company's teaser ads first appeared in the summer of 1977, many of us waited with considerable interest to see what Benton Harbor had wrought. The H-8 was finally unveiled in the autumn, and some were disappointed that Heath had not selected the S-100 bus. Others disliked Heath's choice of split octal notation. Still, the H-8 *was* a Heathkit, which meant you had an excellent chance of assembling it successfully. Now in its fourth year, the H-8 is still a good value.

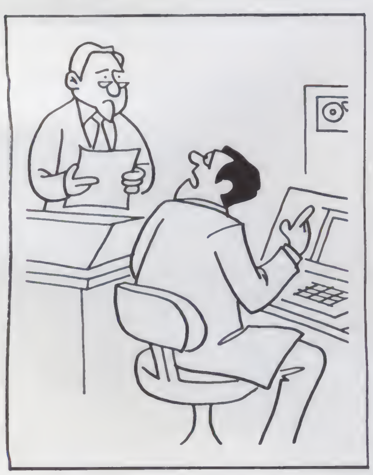
With the All-In-One, Heath has stepped squarely into that market segment dominated by Radio Shack, Apple and Commodore. More importantly, however, they've done this without denying H-89 owners the satisfaction of saying, "I built it myself!"

—The Editors

This month's cover:

The Heathkit H-89 All-In-One Computer. Photo by Martin Paul.

Page 120.



Page 124.

Kilobaud *Microcomputing* (ISSN 0192-4575) is published monthly by Wayne Green, Inc., 80 Pine St., Peterborough NH 03458. Subscription rates in U.S. are $25 for one year and $53 for three years. In Canada: $27 for one year only, U.S. funds. Foreign subscriptions (surface mail)—$35 for one year only, U.S. funds. Foreign air mail subscriptions —$62 for one year only, U.S. funds. Canadian Distributor: Micron Distributing, 409 Queen St. West, Toronto, Ontario, Canada M5V 2A5. In Europe, contact: Monika Nedela, Markstr. 3, D-7778 Markdorf, W. Germany. South African Distributor: KB Microcomputing, PO Box 782815, Sandton, South Africa 2146. Australian Distributor: Electronic Concepts, Attention: Rudi Hoess, 55 Clarence Street, Sidney 2000, Australia. Second-class postage paid at Peterborough NH 03458 and at additional mailing offices. Phone: 603-924-3873.

By Wayne Green

Built-in Computer Serviceman

Self Servicing

One of the things you discover when you computerize any business operation is that computers have a tendency to fail. The second thing you find out is that once the computer stops, your business stops, too. All your employees who were entering data are now on an extended coffee break. All the people who were filling orders generated by the computer are playing games. The office workers who had been word processing are gossiping. A computer breakdown in even a small business can cost you several hundred dollars an hour—plus the aggravation of trying to catch up again once it is back on line, and several hours of work have to be done on overtime.

To give you an idea of what can happen, let me draw upon my own bitter experience. We put in a Prime computer, complete with expensive environmental controls. But things do go wrong. An air conditioner may fail, increasing the temperature a few degrees and thus stopping the whole system for a day or two. We have to protect against heat, cold, dryness, wetness, static, and so forth, with tolerances far more exacting than you'll find in the average business office.

The Prime suddenly stopped working one day. Since we had a maintenance contract, we called the firm for service. The home plant is less than an hour's drive from us, so we expected fairly prompt service—hours, maybe, but a couple of days at the most. Several *months* later, the system was finally back in full working order.

While Radio Shack has been making some positive strides toward providing service for their systems, few other manufacturers have the sales or means to make quick service available.

For a computer to be of any real value to a business, it must work in an office environment, not be sealed in a scientific lab with environmental controls. It should be able to work through chance power outages without faltering. And when something does go wrong, it should be repairable in minutes rather than hours.

There are several alternatives to providing this kind of service. One is to have traveling servicemen who can come to your office to fix the system. This is the system used by IBM, and it works fairly well. When one of my typewriters that is on service contract stops working, I know that I'll have it going within a day or two—except on weekends, which is when my typewriter usually stops. The serviceman has to be able to fix the system on the spot, which is not too difficult for typewriters, but might be more of a problem for computers, which may require sophisticated test equipment. The computer serviceman would probably need a van instead of a car.

In the old TV-repair system, a driver picked up the set, drove it to a service center for repairs and then returned it. This method solves the test equipment and test technician problem, providing more repairs per hour when the tech does not have to drive for an hour to the office site. But this means that your service centers will have to be set up no more than 100 miles apart if you are going to keep trips down to 50 miles or less—which is still at least a two-hour trip for each pickup and delivery. A 100-mile round trip is not going to be inexpensive, adding perhaps $100 to the service call by the time you add in everything. This is not going to be popular.

Another alternative is to have the businessman bring in the computer for repairs. It will cost him a lot less to make that 50-mile trip, and he'll get it done quicker. The computer system is relatively light and easy to move, so this might be a viable alternative. However, since businesses are not used to this, it might not be easy to sell. Repairmen come in to fix their typewriters, copiers and other office equipment, so there might be considerable resistance to having to bring the computer to the lab.

The computer is different from other office equipment in one important way. When a typewriter breaks down, there is no way that it is going to be able to fix itself, or even diagnose its own problems. The office copier, when it gets a piece of paper stuck in its gullet, will not respond to the Heimlich maneuver. It requires an expert to pull that wad out. The computer, on the other hand, can be built and programmed to take its own pulse and let you know what is ailing it.

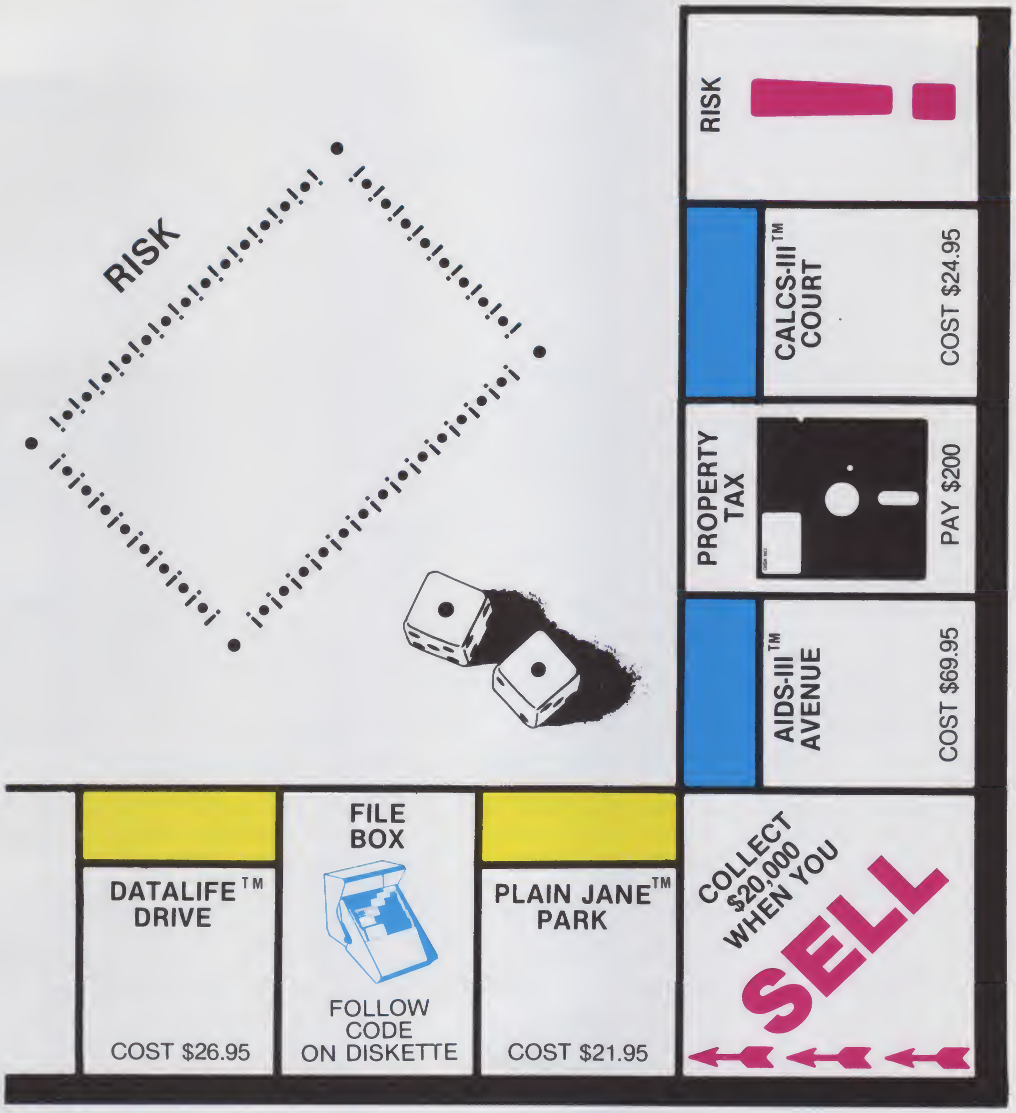
Compugraphic was the first firm that I know of to figure this out. For a bit extra you can buy a service kit along with their computerized typesetter. Thus, when something goes wrong, you pull out the kit and start making simple tests. The kit has repair boards for you to replace if that is called for, and you then send the bummer in by UPS for replacement.

Microcomputers can be built for servicing, with simple-to-reach plug-in boards. Diagnostics that will check for memory errors, CPU problems, power supply problems, etc., can be included with the system. The important thing is to start building the servicing of our systems into the systems. In this way we will be able to provide a far more valuable office tool.

Our office computers should have protection against loss of power. This may come as a battery floating on the line to be used in case of emergency. It might only take the form of a large capacitor which would store enough energy to allow the system to quickly store work in progress for later retrieval.

Perhaps it will be possible for a system of remote diagnostics to be designed which can be done over the phone line via a modem. How service is done is not as important as the speed (high) and the costs (low).□

WHY TAKE A CHANCE WITH YOUR ADVERTISING?



YOU DON'T TREAT YOUR BUSINESS LIKE A GAME . . .

And neither do we. META MEDIA can help you reach your market more effectively and for less money than you may think you have to spend. Business is risky enough. Why not deal with people who understand products for small computers and how to sell them. You have a lot at stake . . . time, money, your future. Why roll the dice?

Meta Media Productions, Inc. 333

26111 Brush Avenue Euclid, Ohio 44132 (216) 289-1100

We Accept: VISA/MasterCard

DATALIFE is a TM of VERBATIM
AIDS-III, CALCS-III and PLAIN JANE are TM of MTC

META MEDIA PRODUCTIONS, INC. is an affiliate of META TECHNOLOGIES CORPORATION, INC.

By Robert W. Baker

Handy Utilities

Make life easier for you and your PET

Pro-Kit #1

This utility package for the PET is a collection of machine code routines that can be permanently located in a protected area at the top of the PET's memory. Once installed, writing programs to accept information from the keyboard is as simple as using INPUT commands, but as safe as a GET routine involving the most comprehensive validation checks. The Pro-Kit routines are called by a few simple POKE and SYS commands. They can easily be incorporated into your own programs, and the authors (Intex Datalog Limited, Eaglescliffe Industrial Estate, Eaglescliffe, Stockton-on-Tees, Cleveland TS16 0PN England) are willing to negotiate terms for using their routines in programs offered for sale.

The simplest routine is the date input routine, which allows inputting a date in a standard format. Only valid dates are accepted; all other inputs are ignored by the routine. It checks for an invalid day for the particular month specified and even handles leap years correctly. The specific format required is a set of three pairs of two-digit numbers representing the day, month and year in that order. This routine is called by a SYS command and returns the date entered as a string in Q$.

A general input routine allows you to specify the maximum length of the entry field and which of the printable characters are to be accepted as valid input. This routine is used for inputting any information that is not a date or a decimal number. To use the routine, you first define a string VC$ that contains each character which the routine will allow as input. Then you poke location 1 with the maximum number of characters you want in the entry field. The routine is called by a SYS command, and the data entered from the keyboard appears in Q$ padded with trailing spaces if necessary.

For numeric input, another routine allows you to specify how many numbers you want before and after the decimal point and whether or not negative numbers are allowed. Before calling this routine you must poke locations 910 and 911 with the number of digits before and after the decimal point. Then poke location 912 with a zero if only positive numbers are allowed, or with a one if positive and negative numbers are allowed. The routine is called by a SYS command, and the user's input value is returned as a properly formatted string in Q$.

The number input routine has a default input value of zero. If you enter a number followed by a decimal point, pressing the decimal point immediately formats the integer part of the field in the proper position. If you enter a small integer number without pressing the decimal point before pressing RETURN, the value returned in Q$ may not be the correct value. This is due to the default characters displayed on the screen in the entry field area. This condition can be avoided by pressing the right-bracket (]) key instead of RETURN. This deletes all characters from the current cursor position to the end of the input field.

If you enter numbers right up to the decimal point, the next number entered automatically goes into the position after the decimal point without pressing the decimal point key. When no numbers are entered prior to the decimal point, Q$ is padded with spaces plus a leading zero. Negative numbers are indicated by pressing the minus sign at any time during the input before pressing RETURN.

Another useful routine allows searching a string in S2$ for a matching substring specified in S1$. A similar search in BASIC could take up to 2.5 seconds, but this routine is almost instantaneous. The routine is called by a SYS command after setting S1$ and S2$ to the appropriate values. Location 32767 then contains a number indicating the starting character position where a match was found in S2$. If the value in 32767 is zero, no match was found.

This routine makes it simple to implement single-key mnemonic responses to a menu without lots of "IF A$ = "XYZ" THEN. . ." statements. Instead, you just define S2$ = "NSEWUD" (or whatever answers are valid) and use a GET statement to receive the option letter in S1$. Then call the PROKIT routine via the correct SYS command and branch to the desired option by a simple "ON PEEK (32767) GOTO . . ." command.

For proper operation of the Pro-Kit routines, the PET must be in the upper/lowercase mode (POKE 59468,14). In this mode, a string in Q$ displayed in uppercase on the screen would appear as either graphics or lowercase on the printer. A graphics conversion routine, called by the proper SYS command, will convert the contents of Q$ to the proper format for being printed as it would appear on the screen.

In all Pro-Kit entry routines, as RETURN is pressed the entry field changes from reverse to normal and a thin line (|) is placed to the right of the entry field. The input routines assume that your entry fields are normally shown with a surrounding box formed using the family of shifted @[]123+0 characters.

Another function common to all Pro-Kit entry routines is the action caused by pressing the RVS key instead of RETURN. This acts just like a RETURN, but location 32767 is set to a one instead of a zero. The appearance of a one in this location can then be used as an indication that something special should happen in the program after the entry is made. For example, it could mean to step back to the previous input instead of continuing to the next expected input.

Another three routines provide various screen-exchange functions. One routine takes whatever is displayed on the screen and dumps it in a specified area of memory. The second does the reverse, bringing back an area of memory to be displayed on the screen. The last routine exchanges an area of memory with the screen display. Before calling this routine by the SYS command, you must poke six memory locations. Two locations specify

IF YOU CAN WAIT A MINUTE, WE CAN SAVE YOU $1,000.

With the Starwriter™ Daisy Wheel 25 cps printer from C. Itoh.

A business letter, written on a 45 cps word-processing printer, might take about two minutes to print.

With the Starwriter, it might take closer to three.

The typical 45 cps printer retails for about $3,000.

But the Starwriter 25 retails for about $1,895—thus saving you about $1,000.

And therein lies the biggest difference between the Starwriter 25 and the more expensive, daisy wheel printers.

The Starwriter 25 comes complete and ready-to-use, requiring no changes in hardware or software. It uses industry-standard ribbon cartridges, and it's "plug-in" compatible to interface with a wide variety of systems, to help lower system-integration costs.

Using a 96-character wheel, it produces excellent letter-quality printing on three sharp copies with up to 163 columns, and offers the most precise character-placement available, for outstanding print performance.

C. Itoh's warranty;

3 months on parts and labor, supported by one of the best service organizations in the industry.

$1,000 OFF

Leading Edge Products, Inc.,
225 Turnpike Street,
Canton, Massachusetts 02021

Dear Leading Edge:
I'd like to know more about the Starwriter, and how spending a minute can save me a grand.
Please send me the name of my nearest dealer.

Name____________________
Title____________________
Company____________________
Street____________________
State____________ Zip__________
Phone: Area Code____________________
Number____________________

LEADING EDGE. 355

Leading Edge Products, Inc., 225 Turnpike Street, Canton, Massachusetts 02021
Dealer inquiries invited. For immediate delivery from the Leading Edge Inventory Bank™ call toll free 1-800-343-6833
In Massachusetts, call collect (617) 828-8150. Telex 951-624

the start of the memory area to be used while the other four locations determine which of the three screen exchange functions is to be performed. It is the user's responsibility to change the BASIC top-of-memory pointers to protect a suitable area of memory for the desired application.

The Pro-Kit routines can be easily incorporated into your own programs. However, the suggested method of saving a program with Pro-Kit is to save all RAM using the monitor. Thus, each program will be 32K long and take about 125 blocks on disk. You can save just the program as normal without Pro-Kit, but then you have to load Pro-Kit separately each time the program is loaded.

This package of routines is handy and very impressive, but the assumed formats may or may not be what you would like. The printed documentation is clear and complete, with various exercises to explain the operation of each routine. The programs are available on disk or cassette with additional demo programs included. They're all written for the 3032 PET (a 2001 with BASIC 3.0) and will not run with the BASIC 4.0 ROMs or on the 8016/8032 machines.

Utility Programs

I recently received a collection of various utility programs for review from California Software Associates (Box 969, Laguna Beach, CA 92652). I've included information on two of the programs I was able to try.

EDIT is a BASIC program that allows editing or appending to sequential or user diskette files on the 2040 disk. The edit function acts on files in place on the diskette, so records are limited to the original record length. When appending records to an existing file, the new records are limited to 80 characters per record. In either mode, colons and commas are ignored.

When the program is run you are asked for the desired filename and the disk drive number. The program then locates and traces the file chaining on the disk by track and sector. The number of blocks in the file and the number of free blocks on the disk are both displayed. For large files this can take some time (3.5 minutes, worst case), but it only occurs once. All following actions should be virtually instantaneous.

After all the initial setup is complete, you are asked whether you want to EDIT, APPEND or STOP. If you accidentally choose STOP, a RETURN will reenter the program. When you select EDIT you're asked for the record number to be edited. You can enter the exact record number, an exclamation point or RETURN to exit edit mode.

After entering a record number between one and the maximum record number, the program displays the current contents of the record with an up-arrow under the first character position. Once you position the up-arrow to the appropriate starting position, you simply type in the desired edit. Inserts into a full record delete any trailing characters, while deletes append trailing spaces. If you try to type beyond the end of the record, the last character of the record equals the last character typed. There is no wrap-around, and you cannot change the size of the record. When satisfied with the changes, hitting RETURN causes the new record to replace the original record in the file.

If you respond to the edit record number prompt with an exclamation mark, you are then asked for a search string of up to 80 characters in length. Leading question marks in your search string function as "don't care" or "wild card" characters. No character matching is performed in these positions. A sequential search of the file is performed, looking for a match but only up to the length of the search string specified. Any trailing characters in the file record are ignored.

If the search is successful, editing proceeds as outlined previously. Otherwise, if the search fails, you will be asked again for an edit record number. Searching through long files can be rather slow. If you know about where the record is, using several record number guesses is much faster.

When appending new records, you are prompted for each new record with a maximum length of 80 characters. The delete key functions as normal while the cursor movement keys are disabled. You can add records to the limit of the block availability on the diskette. The appended records can later be edited, but they cannot be deleted once entered.

SNAPSHOT is a machine-language screen utility program that allows saving and restoring up to ten different copies of the CRT display. On the 2001 series PETs the program occupies the first and second cassette buffers, so neither cassette may be used while it is in place. It is normally compatible with BASIC as well as most other utilities such as WEDGE/DOS, TOOLKIT, EXTRAMON, etc. The program will usually notify you and abort if there is a utility in place with which it is not compatible. Various versions are available for each model PET/CBM and for the different ROM sets.

SNAPSHOT is loaded and run after all other utilities are loaded. It first asks for the maximum number of pages to be saved. If there isn't enough memory available, an error message is displayed and you get to try a smaller number. Otherwise, the program initializes itself and displays the number of bytes remaining for BASIC along with simple calling procedures for the program.

SNAPSHOT can only be called from the upper-left corner, or cursor-home position, of the screen. The first character in the call must be either an exclamation mark or an equal sign. The second character must be an allowable page number or an "X" (to deactivate SNAPSHOT). Each call is terminated with a RETURN, and any other data on the first line is ignored. An exclamation mark is used to copy the current screen contents to the section of RAM reserved for the indicated page number. An equal sign is used to recall a snapshot of the screen. The section of RAM reserved for the indicated page number is copied back to the screen.

Any SNAPSHOT call leaves the cursor in the cursor-home position so you can easily cycle through current snapshots of the screen. Once a snapshot has been taken, control is returned to BASIC command mode in the same condition as prior to the SNAPSHOT call. Thus, im-

mediate commands can be executed, programs run or continued, etc.

Although SNAPSHOT saves and recalls images of the PET screen, it does not save any line-wrapping (linking) information when the snapshot is taken. Similarly, SNAPSHOT does not change the line wrapping (linking) in effect before a snapshot is recalled. This can cause unpredictable or undesirable results if certain precautions are not taken if you intend to use any snapshot line as an immediate BASIC command or a program line. When SNAPSHOT is deactivated by an "!X" call, the BASIC CHRGOT routine and the IRQ vector in low memory are restored to the BASIC cold-start condition. This may deactivate other utilities, but BASIC is left untouched.

I found both programs to be very handy and encountered no problems with either. The EDIT program can be purchased separately for $60 or as part of a complete disk utility package for $100. The utility package also includes:

MASSAGE—which allows editing disk blocks, updating the disk block availability map (BAM), scanning files, mapping the disk, tracing block linkage, etc.

RECOVER—which recovers scratched files.

SNAPSHOT sells for $25, and there are different versions for the various models and ROM sets as mentioned. On all programs sold by California Software Associates, there is an update service available.

Word Pro Disk Files

With the many valuable features of Word Pro, it's not surprising that more and more programs are using Word Pro to create input data or utilize their output data. In fact, there's no real reason why your own programs can't interface to Word Pro as well. It's really quite simple once you know the disk file format and character set used by Word Pro.

All disk files saved or read by Word Pro are actually program files with a load address as the first two bytes of the file. With Word Pro 3 the actual address is 5108 hex and is stored in the normal 6502 address format (low byte first—08; high byte last—51). The file then contains the actual text with any embedded control commands stored exactly as displayed on the screen. Each line is stored as 40 bytes regardless of its contents; there is no compacting of the data. If the only thing on the line was a RETURN (stored as a back-arrow), it would be stored as the RETURN followed by 39 spaces. If you are using Word Pro 4, then remember that there will be 80 characters per line instead of 40.

Now for the tricky part! The value stored for each character in the file is the same value as used in the screen RAM when the PET is in the upper/lowercase mode (POKE 59468,14). These are the same values you would normally use when poking values directly into the display. Remember that the value used to generate a specific character can be different depending on the character set that is active—upper/lowercase or uppercase graphics. Since Word Pro 3 forces the upper/lowercase mode, we'll only be concerned with that character set.

Since Word Pro uses the OFF/RVS key to enter control mode, you cannot generate reverse characters. This limits the character set used by Word Pro to only 128 characters. There are really only 123 usable characters, since five characters are reserved for special purposes (RETURN, begin/end underline or enhance, tab, toggle uppercase lock). Table 1 shows the decimal and hex value of each character as stored/read by Word Pro.

If you want to read a file created by Word Pro (or by another program in Word Pro format), simply open the file as a sequential read file (OPEN 1,8,1,"0:filename,S,R). Read and ignore the first two characters, which represent the load address for the program file. Then read the data using a GET#1,C$ (or similar) command and translate the data into the normal character set if necessary.

To write a file for Word Pro or in the Word Pro format, first open the file (OPEN 1,8,1,"0:filname,P,W). Now write the normal Word Pro load address: PRINT #1,CHR$(8);CHR$(81); followed by any desired data. Remember to write the correct value for each character and always use the ending semicolon on the print statement. This will write individual bytes of data in the format desired.

Miscellany

Espon's new MX-80 printer offers a multitude of features normally unexpected in a low-cost printer. With an available IEEE-488 interface and an optional graphics ROM, it can easily be used with the PET. If you should decide to buy an MX-80, however, be sure you get the correct interface module. There were apparently two IEEE-488 interface boards—a model 8160 and an 8161. Only the 8161 board will work correctly with Word Pro. The original 8160 board will not handle all PET printing modes correctly.

If your company or organization is offering products of interest to PET/CBM owners, I'll be happy to review them here as space permits. Reviews are generally done in the order received. If you'd like something reviewed and timed to appear with a product announcement, please remember the typical three-month lead time for this column. As always, any and all correspondence is welcomed but should be addressed to me at my home address (15 Windsor Drive, Atco, NJ 08004). An SASE is appreciated if you expect a response.□

Dec	Hex	Char	**	Dec	Hex	Char	**	Dec	Hex	Char	**	Dec	Hex	Char
0	00	@		32	20	space		64	40	shift-@		96	60	shift-spc
1	01	a		33	21	!		65	41	A		97	61	▌
2	02	b		34	22	"		66	42	B		98	62	▄
3	03	c		35	23	#		67	43	C		99	63	‾
4	04	d		36	24	$		68	44	D		100	64	_
5	05	e		37	25	%		69	45	E		101	65	▏
6	06	f		38	26	&		70	46	F		102	66	▒
7	07	g		39	27	'		71	47	G		103	67	‾
8	08	h		40	28	(		72	48	H		104	68	▒
9	09	i		41	29	)		73	49	I		105	69	◤
10	0A	j		42	2A	*		74	4A	J		106	6A	▕
11	0B	k		43	2B	+		75	4B	K		107	6B	├
12	0C	l		44	2C	,		76	4C	L		108	6C	shift-,
13	0D	m		45	2D	-		77	4D	M		109	6D	└
14	0E	n		46	2E	.		78	4E	N		110	6E	┐
15	0F	o		47	2F	/		79	4F	O		111	6F	shift-/
16	10	p		48	30	0		80	50	P		112	70	┌
17	11	q		49	31	1		81	51	Q		113	71	┴
18	12	r		50	32	2		82	52	R		114	72	┬
19	13	s		51	33	3		83	53	S		115	73	┤
20	14	t		52	34	4		84	54	T		116	74	▏
21	15	u		53	35	5		85	55	U		117	75	▎
22	16	v		54	36	6		86	56	V		118	76	/
23	17	w		55	37	7		87	57	W		119	77	‾
24	18	x		56	38	8		88	58	X		120	78	▬
25	19	y		57	39	9		89	59	Y		121	79	▄
26	1A	z		58	3A	:		90	5A	Z		122	7A	(wp-cmd)
27	1B	[		59	3B	;		91	5B	+		123	7B	shift-;
28	1C	(up-case)		60	3C	<		92	5C	▒		124	7C	(beg-ul)
29	1D	]		61	3D	=		93	5D	|		125	7D	┘
30	1E	|		62	3E	>		94	5E	▒		126	7E	(end-ul)
31	1F	(tab)		63	3F	?		95	5F	shift-?		127	7F	▚

Table 1. Decimal and hex values of Word Pro characters. The commands are indicated by a value of 122 (7A hex) in the first column of any display line. The command then follows with each character in the command as shown in the table. The uppercase lock is enabled/disabled by the value of 28 (1C hex). A value of 31 (1F hex) indicates a tab. The values of 124 (7C hex) and 126 (7E hex) begin and end underlining or enhancing, respectively.

By Walter Koetke

First Impressions Are Forever

Get off to a good start with computers

First-Day Success

First impressions are important whenever you meet someone. First impressions are also important whenever someone meets a computer. Every effort should be made to make the first few experiences of the new computer user enjoyable. An enjoyable, successful first experience for students can go a long way toward making their future attitude regarding computer use a positive one. A successful first experience for teachers is almost a necessity if you want them to continue to explore the educational potential and reality of microcomputers in their classrooms.

During the late 60s and early 70s many interested, capable teachers were nearly, if not completely, convinced that computers could have little impact on classroom activities. These teachers had heard about time-sharing terminals connected to minicomputers and perhaps seen a few enticing demonstrations. Their curiosity was aroused, so they enrolled in an introductory computing course in a local college or university.

That was the beginning of their decline. Many were taught to keypunch in preparation for batch processing. After all, the local college didn't yet have time-sharing terminals, "but the principles are all the same." Many others were taught FORTRAN, COBOL or RPG, because that's what the local college staff was prepared to teach. And still others were taught bits, bytes and assembly language because higher-level languages were less prestigious.

For the teacher seeking to explore classroom applications of computing facilities, these courses were, and still are, a disaster. They left the course convinced that computing, as they had experienced it, was of no use in their classroom. And they were right! Most unfortunately, those aspects that were appropriate for the classroom remained beyond their experience, and their now negative attitude was likely to keep it there.

Don't let a similar scenario adversely affect the positive potential of the microcomputer. College and other courses that use other than microcomputer hardware, that do not teach beginners BASIC or that do not provide hands-on experience are very likely doing more harm than good when the courses are billed as appropriate introductions for educators. The educational potential of the $1000 microcomputer far exceeds that of other forms of computer support that is realistically available to today's elementary and secondary schools. If teachers are willing to take courses to improve their skills, they should not be given inappropriate material because that's the only thing available. Teachers should seek those courses that will provide the introduction they need, and must not settle for less.

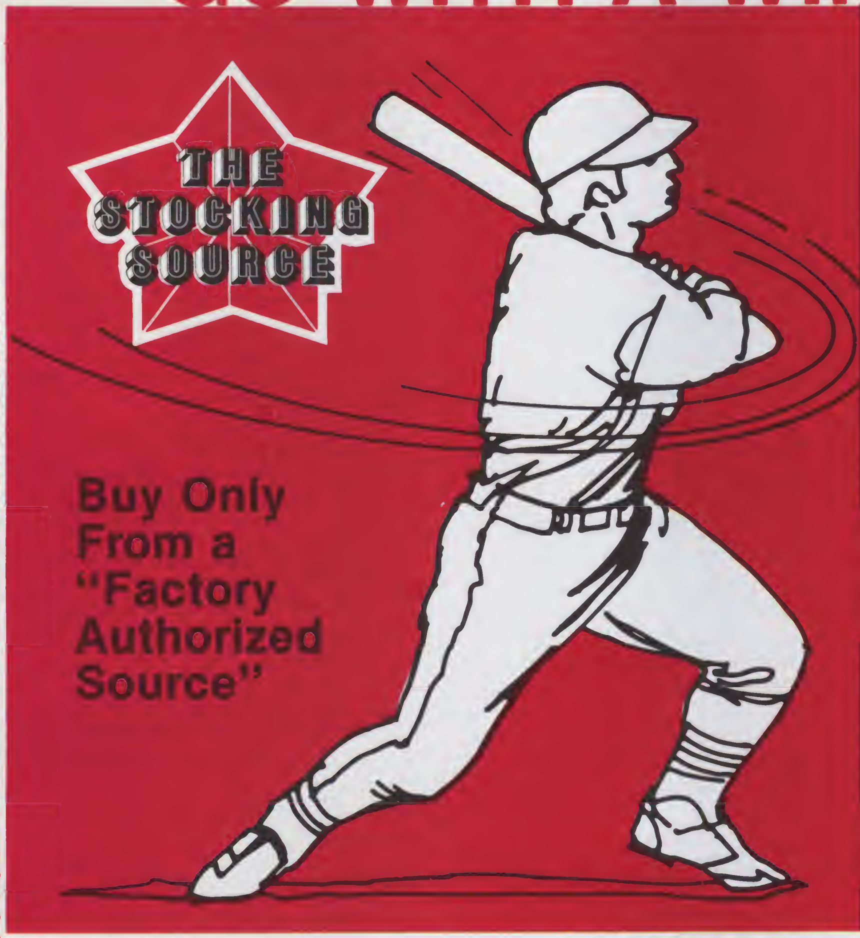
On a much smaller scale, teachers should strive to make a student's first few computer experiences positive and rewarding. One very common way to do this is to allow students to begin by playing one or two popular games. This is only acceptable if the game is carefully chosen; it must be user-proof. The inexperienced user should be able to type absolutely anything without seeing an error message. The game must not be invincible. A first-time user should be more likely to win than lose. There is little reward in having a computer beat you in four out of four games, especially if they're the only four games you've ever played with the computer. The game must be exceptionally easy to understand, and the game should be fun.

One program that meets these qualifications asks the user to guess a secret integer between 1 and 100. Each incorrect guess is indicated as too high or too low. After the number is guessed correctly, the roles are reversed and the computer guesses a number made up by the user. Listing 1 contains a TRS-80 version of

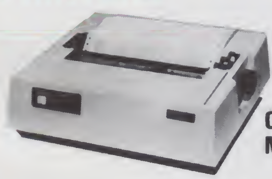
```
100 CLS : PRINT @256, "I'M THINKING OF AN INTEGER BETWEEN 1 AND 100 INCLUSIVE"
110 PRINT : PRINT "TRY TO GUESS MY INTEGER"
120 CLEAR 200 : L=0 : H=101 : N=RND(100) : GOSUB 400
130 CLS : PRINT @256,"YOUR GUESS "; : INPUT G$ : G=VAL(G$)
140 IF G<=0 OR G<>INT(G) OR G>100 THEN PRINT "PLEASE USE INTEGERS BETWEEN
    1 AND 100" : GOSUB 400 : GOTO 130
150 IF G<=L THEN PRINT "YOU ALREADY KNOW MY NUMBER IS HIGHER THAN" L :
     GOTO 200
160 IF G>=H THEN PRINT "YOU ALREADY KNOW MY NUMBER IS LOWER THAN" H :
    GOTO 200
170 IF G<N THEN PRINT "TOO LOW" ; L=G
180 IF G>N THEN PRINT "TOO HIGH" :  H=G
190 IF G=N THEN PRINT @256, CHR$(23)+"YOU GOT IT!   GOOD WORK!"
200 GOSUB 400 : IF G<>N THEN GOTO 130
210 CLS : PRINT @256, "NOW WE'LL REVERSE THE PROCESS"
220 PRINT @384, "YOU THINK OF AN INTEGER BETWEEN 1 AND 100 INCLUSIVE,"
230 PRINT "AND I'LL GUESS IT"
240 L=0 : H=101 : L1=1 : H1=100 : GOSUB 400 : GOSUB 400
250 G=INT((L+H)/2)
260 CLS : PRINT @256, "I GUESS" G : PRINT
270 INPUT "AM I (H)IGH, (L)OW, OR (R)IGHT"; R$
280 R$=LEFT$(R$,1):IF R$<>"H" AND R$<>"L" AND R$<>"R" THEN GOTO 270
290 IF R$="H" THEN H1=H : H=G
300 IF R$="L" THEN L1=L : L=G
310 GOSUB 400 : IF G=L1 OR G=H1 THEN CLS : PRINT @256, "YOU'RE NOT GIVING
    ME CONSISTENT ANSWERS" : PRINT "LET'S TRY THIS PART AGAIN" : GOSUB
    400 : CLS : GOTO 220
320 IF R$<>"R" THEN 250
330 CLS : PRINT @256, CHR$(23)+"THAT WAS FUN  -- I LOVE NUMBERS!" : GOSUB
    400
340 CLS : PRINT @256, "HOPE YOU ENJOYED THIS EXERCISE" : PRINT
350 PRINT "I'LL BET YOU WRITE BETTER ONES PRETTY SOON"
360 GOSUB 400 : GOSUB 400 : END
400 FOR I=1 TO 1000 : NEXT I : RETURN
```

Listing 1. Number-guessing game for the TRS-80.

GO WITH A WINNER!



PRINTERS



Okidata Microline 82

	List Price	Your Cost
NEW — Okidata Microline 82	$ 960.	$800 Ask For Our Price
Epson Model MX-70	$ 495.	Ask For Our Price

Features: 80 C.P.S.; Bidirectional; Full forms Control; Choice of Parallel or Serial Interface as Standard; Buffered Serial Optional

Epson MX-70 Features: 80 C.P.S.; Unidirectional; Bit Plot Graphics; Adjustable Sprocket from 4" — 10"

	List Price	Your Cost
Epson Model MX-80	$ 645.	Ask For Our Price
Okidata Microline 80	$ 800.	$550 Ask For Our Price
Anadex DP-9500 & DP-9501	$1650.	Ask For Our Price
Anadex DP-8000	$1095.	$895 Ask For Our Price

INTERFACES —

Epson Apple Plug-in Interface & Cable . . . . . . $110.
Epson Serial Interface & Cable . . . . . . $ 90.
Epson IEEE 488 Interface & Cable . . . . . . $ 80.
Model TRS-80 Expansion Interface Cable . . . . $ 35.
Model 232 Universal CA Cable, Universal Serial Interface . . . . $ 25.
Okidata Microline 80 Tractor . . . . $100.
Okidata Microline 80 RS-232 Interface with 256 Character Buffer . . . . $170.

SYSTEMS FURNITURE — We stock Systems Furniture tables.

PRINTER STANDS — We stock Universal printer stands for most printers.

COUPLERS & MODEMS — We handle the broad Tek-Com line.

BREAKOUT BOX

	List Price	Your Cost
NEW — HAYES TECHNICAL	$195	Ask For Our Price

Full RS 232 Breakout and Patching Interface Box

master charge VISA

ASK FOR OUR INSTANT DISCOUNT From Roy Hawthorne
Talk To Bill Tokar On Applications

CALL TOLL FREE
U.S.A. 1-800-521-2764
MICHIGAN 1-800-482-8393

Reminder: We are open 8:30 AM to 5:00 PM EST Monday through Friday

WRITE TO:
"The Stocking Source" ✓288
NEW 24069 Research Drive
Farmington Hills, MI 48024
313-474-6708

this program. You'll have to type the program to see a sample run.

Notice that the program attempts to deal with all possible incorrect user responses by providing helpful messages. The user should never be faced with a BASIC error message or a prematurely terminated program. Note that line 130 uses a string variable when a numeric value is expected. This allows the user to enter anything without seeing the ?REDO message. Also notice that large letters are used as a change-of-pace reward. While this is a very simple example, programs should make the reward worth the effort. Finally, notice that the program doesn't allow the user to lose nor does it evaluate his performance. Although both of these elements are often desirable, neither is necessary in the student's first experience with the computer.

After the successful completion of a game or other interactive program, the first-time user can be introduced to programming. Although the first programs will not be elegant, they should be fun. The following sequence of first-encounter programs is suggested because they introduce new commands and elementary programming concepts in a way that is easy, fun and likely to lead to many other interesting questions.

Have the students enter and run the following initial program:

```
10 PRINT "JULIE"
20 GOTO 10
```

They should, of course, enter their own first names. This allows the introduction of line numbers, three commands (counting RUN), strings and the break key. Since almost everyone wants to see their name in lights, the program is fun.

The second program is a modification of the first. The string in line 10 should contain two spaces after the name and then be followed by a semicolon. This program appears as:

```
10 PRINT "JULIE  ";
20 GOTO 10
```

Execution of this program always brings delight. And when done in an area with several microcomputers available, the effect is greatly enhanced. On the TRS-80, the program will fill the screen with JULIE and appear to be moving to the left. However, the name TOM will move to the right, while the name WALTER will produce relatively dull columns that aren't moving at all.

Why do some names move left, some right, and some not at all? I've asked these questions of first-time users from third grade through adults who have answered correctly in each instance. Even third-grade students correctly answer the question. Although they haven't had formal instruction about division, they can correctly explain the results of this program by describing the division process and the remainder in their own words.

Another program for those learning to program is:

```
10 LET N=1
20 PRINT N
30 LET N=N+1
40 GOTO 20
```

This program is an effective vehicle for introducing the LET command. Ask students how to begin counting at 1000 rather than 1. Ask them how to count by 2's, by 5's, or how to count down.

Also, ask students to determine approximately how much time the computer will take to get to one million. After all, a device that measures time in microseconds should certainly be able to count from one to one million very rapidly.

The results of this assignment are almost always surprising (a TRS-80 requires about 8-1/2 hours to count from one to one million in this way), and the students' own activities as they develop procedures for obtaining their results are quite valuable. There are several related points that should be discussed. The most significant of these, in my opinion, is the largeness of large numbers. Very few children or adults have any intuitive feeling regarding large numbers. Most people watching this program being executed will guess that one million will be reached in two hours or less.

One million is a lot larger than we think. When the news reports that a new plane costs eight million dollars, someone is spending a lot of money. When the national debt exceeds one trillion dollars, we know that's a large number. But we have little feeling for how very large it really is. How long would it take our short program to count from one to one trillion?

```
100 L$="ABOVE" : H$="ZEBRA" : C=0
110 FOR X=1 TO 800 : NEXT X : CLS
120 PRINT @256, "TYPE A 5 LETTER WORD THAT IS ALPHABETICALLY" : PRINT
     "BETWEEN  '" L$ "'  AND  '" H$ "' ";
130 PRINT @768, C " ENTERED SO FAR"; : PRINT @349, ; : INPUT N$
140 IF LEN(N$)<>5 OR L$>=N$ OR H$<=N$ THEN PRINT @448, "NO, TRY AGAIN"
     : GOTO 110
150 C=C+1 : IF C=15 THEN 200
160 FOR I=1 TO 5
170 N=ASC(MID$(N$,I,1))
180 A=N-ASC(MID$(L$,I,1))
190 B=ASC(MID$(H$,I,1))-N
200 IF A<B THEN H$=N$ : GOTO 110
210 IF B<A THEN L$=N$ : GOTO 110
220 NEXT I
```

Listing 2. Word Squeeze program.

Another topic often occurs when the more advanced student points out that he added the line 40 IF N<10000 THEN 20 to make it easier to time the program. That this command also consumes time during each loop should not be overlooked. How much time? I'll leave that as an exercise for the reader.

Secondary students might also calculate the time required for the following program to count from one to one million.

```
10 LET N=1
20 LET N=N+1
30 GOTO 20
```

Consider one more example. This one is for the many teachers and students whose first computer experience should not include numbers if enjoyment is an objective. The dependence upon sample programs that draw heavily on mathematical examples is not at all necessary. The TRS-80 program given in Listing 2 is a bare-bones version of a rather enjoyable game that we'll call Word Squeeze.

Word Squeeze begins by asking you to type a five-letter word that is alphabetically between "above" and "zebra." After you type your word, the program determines the shortest alphabetic "distance"—above to your word or your word to zebra. Suppose you typed the word "class." The alphabetic distance from above to class is shorter than that from class to zebra.

After the program determines which words define the shorter distance, the user must enter a word that is alphabetically between them. If, for example, you typed class, then you would next be asked to type a word that is alphabetically between above and class. If you responded with the word "break," you would next be asked to type a word that is alphabetically between break and class. And on and on.

In the Listing 2 program, the user does not win or lose Word Squeeze, he just plays. The program is offered as a starting point to which you can add many bells and whistles of your own. You might, for example, permit words of three or more letters rather than only using five-letter words. You might also check that each word entered contains at least one vowel and at least one consonant. Certainly, you can improve the scoring procedure, and you ought to provide a method for the user to quit more gracefully than pressing break.

Make every effort to make the first experiences of new users enjoyable ones. The ideas offered in this article hopefully demonstrate that these experiences can be not only enjoyable, but also challenging and stimulating. If you've some similar ideas that have worked for new users, send them in. Perhaps the best ones will be appropriate for a future column. □

Address correspondence to Walter Koetke, Putnam/Northern Westchester BOCES, Yorktown Heights, NY 10598.

New muscle for Heathkit Computers

Now with new CP/M® Systems Software for thousands more programs...and new 8-inch disk system for millions more bytes.

The Heathkit All-In-One Computer now has space for 64K of addressable RAM, so you can run bigger, more complex programs.

And our new CP/M® Operating System (Standard *ORG-0, Version 2.2*) makes thousands of CP/M programs available to you. Heathkit systems can run more, do more, store more than ever before.

A new 8-inch double-sided, double-density disk system, with over 1 million bytes per drive, is now available for Heathkit H-89 and H-8 Computers.

The new 8-inch disk system features soft-sectored disks for IBM® compatibility. It's capable of operating in standard IBM 3740 format. And the 8-inch system can be used in conjunction with 5¼-inch systems. **For compatibility with the rest of the industry, Heathkit Computers may just be the most flexible systems you can buy.**

All Heathkit computers and peripherals are available in money-saving, easy-to-build kit form — or completely assembled and factory tested. All are supported by the best documentation in the business and by 62 service centers throughout the U.S. and Canada. You're never out in the cold.

For complete details and prices on Heathkit computers, peripherals and software, write today for the latest Heathkit Catalog or visit your nearby Heathkit Electronic Center*.

VISIT YOUR HEATHKIT STORE

Heathkit products are displayed, sold and serviced at Heathkit Electronic Centers*, located in major cities throughout the U.S. and Canada. See your white pages for the store nearest you.

*Units of Veritechnology Electronics Corporation, in the U.S.

Send for FREE catalog

Write to Heath Company, Dept. 351-754, Benton Harbor, MI 49022

In Canada, contact Heath Co., 1480 Dundas St. E., Mississauga, Ontario L4X 2R7

Heathkit® ~236

IBM is a registered trademark of International Business Machines Corporation. CP/M is a trademark of Digital Research, Inc.

CP-196

LETTERS TO THE EDITOR

Warning for Superboard II Users

If you own an Ohio Scientific Superboard II microcomputer and your power supply output is limited to 3 amps (such as a homebrew LM323K circuit), check the Superboard's fuse F1. I recently received my Superboard and hooked up an overvoltage protect crowbar circuit by clipping the crowbar's +5 V lead into the fuse clip (along with the fuse end) that is connected to the load side of the fuse (Superboard's +5 V bus). The crowbar was initially set to fire at too low a voltage, and fired one day when I was using the computer. However, fuse F1 did not blow, resulting in an extremely hot LM323.

The reason? As supplied, F1 was a 5 amp, 250 V fast blow fuse. The LM323's output current is limited to 3 amps, and so couldn't blow the 5 amp fuse. I was fortunate, as my crowbar circuit caused the short and I was right there to check it out. The real danger is a shorted IC on the Superboard, with no one around. The LM323 will just keep feeding 3 amps into the short until power is removed or something literally burns up, probably destroying the board.

I read in an article titled "Build a Home for Your Superboard" (April 1980 *Microcomputing*, from where I also got the crowbar circuit) that an 8K Superboard draws about 1.8 amps. Therefore, a 2 to 2.5 amp, fast blow fuse will hold, but definitely not larger than a 3 amp if you're using a 3 amp supply. I have a 4K Superboard and am using a 2 amp fast blow.

Ron Hassinger
RAF Chicksands, England

Paradoxical Processing

Your comments about word processing in the November issue ("Word Processor Woes," p. 7) touched one of my nerves, resulting in this letter. Several months ago my office took delivery of an NBI 3000 Word Processing System. This electronic phenomenon has produced more paradoxes than I possibly could have foreseen. For example:

The system produces documents of unbelievable quality, mostly due, however, to the excellent Diablo printer.

The turn-around time for a normal two-page proposal letter, from dictation to final document, has not decreased, but rather has almost doubled.

The freedom associated with the easy editing and correct capabilities has resulted in more relaxed and productive dictation and transcription sessions.

Although well-written, the operator's guide serves more as a reference manual, rather than as a training guide. Consequently, finding how to do something that isn't done very often is difficult and time consuming.

Every letter now goes through a "draft" stage, whereas many letters used to get typed only once, and then signed and mailed.

Many difficult typewriter functions have been simplified to a few keystrokes on the word processor. This, however, is offset by the fact that many easy typewriter functions, such as the "hanging indents," have become more complicated or time consuming.

The word processor doesn't replace a typewriter, but is still needed for many pre-printed forms or short notes.

The confounded machine picks the absolutely worst times to fail. It always happens at 3 PM as the almost-final draft of a twelve-page document (that must be mailed today) is being printed.

While the list could go on, I think it sufficiently expresses my frustrations. I am pleased with the professional quality that word processing lends to my correspondence and documents. I like the flexibility that word processing offers. I think the per-station costs are too high, but these will come down sharply over the next couple of years. I have a difficult time accepting the fact that while some tasks became easier, many other, more often used tasks became more cumbersome. But, no matter how well the letter is printed, regardless of the technology used to produce the document, in spite of the time and effort invested, the Post Office will still manage to lose it, mis-deliver it or destroy it!

Roland K. Smith
Datapoint Eastern Region
Sales Manager
Cleveland, OH

Megabytes on Cassette

A product has been on the market now for about a year that does just what you asked for in your November 1980 "Publisher's Remarks" (New Ideas, p. 8)—that is, allow the storage of 10 megabytes of Winchester data on a regular cassette.

The Corvus Mirror is a low-cost backup for the Corvus Winchester disk system employing standard video recording technology. It converts the output of the Corvus 10 or 20 megabyte Winchester disk into a stream of serial bytes which can be recorded on any videotape recorder. The most common video machine used for this purpose is the video cassette recorder because if offers the convenience and low cost of easily available cassettes.

Data is backed up at a one megabyte per minute rate, permitting the entire contents of a Corvus Winchester disk to be copied in 10 to 20 minutes. Total capacity of a cassette is 120 megabytes. Thus, the Mirror serves not only as a security backup, but also provides archival storage. In a larger sense, it allows microcomputer users to enjoy both the sealed environment reliability of the Winchester disk system and the convenience of cassette loading at the same time by reloading cassette data back onto the disk as needed.

Joseph D. Hughes
Vice-president, Marketing
Corvus Systems, Inc.
San Jose, CA

A Change in the Standings

Here's some new information that changes some of the data in your recent article "The 16-Bit Time Trials" (*Kilobaud Microcomputing*, October 1980, p. 182).

The Z8000 CPUs are now available from both Zilog and Advanced Micro Devices in 6 MHz clock rate versions (called the 8001A and 8002A). Use of the A version reduces cycle times to two-thirds that of the 4 MHz parts, resulting in execution times and rankings in your Table 2 of 32 us (#1) for table lookup, 712 us (#3) for block move, 5.3 us (#1, tie) for jump table and 13.3 us (#3) for multiply. The execution time index changes to 0.596 (#2), and the overall index changes to 0.799, which makes the Z8000 number one in rank.

John Springer
Advanced Micro Devices, Inc.
Sunnyvale, CA

All Shapes and Sizes

Arnold Bragg's article "Area Estimation" (October 1980, p. 112) showed an amazingly effective algorithm for its simplicity. Its greatest weakness is that only straight lines (not curves) are fitted between the data points.

To test the formula for its ability to fit curves, I used a circle with the radius of 1 and chose the data points so they were equally spaced around the circumference. For comparison, I also calculated the area of an N-sided inscribed polygon. The area determined by the formula is always the area of the inscribed polygon.

Fitting a curved line is more of an inconvenience rather than a fault of the al-

gorithm. Any inconvenience the fitting creates is more than made up for by the algorithm's ability to fit very complex shapes. The only time the algorithm failed was for the ring figure, and I tried to do the outside and the inside area in one pass. This could easily be handled by making two passes—one for the outside and the second one for the inside—and then subtracting the differences.

Bob Fruit
Hinsdale, IL

A Sound Idea

It seems to me that it would be fairly easy to rig up one of the popular microcomputers to print stenotype which puts the sounds of words on paper, from voice. You would need something to recognize phonemes (the Radio Shack Vox Box might do) and a special character generator of stenotype symbols for the video display and/or printer. The software would amount to a lookup table to make the proper conversion.

Three of the four major television networks are now captioning some of their programs for the deaf. But captioning costs about $2500 per hour. If the deaf can learn to read stenotype, then the captions could be produced automatically.

A deaf person with his own steno converter wouldn't need broadcast captions. He wouldn't even be limited to television; he could "hear" the radio, the telephone and people in the room too. When all this is boiled down into a handheld unit—the size of the TRS-80 Pocket Computer, say—he could use it wherever he is, eliminating much of his disability.

The hardware for a home system is already available, or nearly so. The software should be easy. There should be a reasonable market for it (don't forget secretaries typing from recordings). Any takers?

Everett Ogden
Delmar, NY

Inflation

I've followed you for the last 25 years through many troubled times, and I always enjoy your editorials. That is, till October 1980 in *Kilobaud Microcomputing* ("The Detroit Syndrome," p. 6).

I am a tool and die maker and a UAW (United Auto Worker) member. I subscribe to three of your magazines—*73*, *Kilobaud Microcomputing* and *80 Microcomputing*. I make my living as an overpaid auto worker, and I buy an overpriced magazine. I work hard and with good productivity. My company (J. I. Case) makes a good profit.

Robert L. Kennedy K9DXB
Merom, IN

By Frank J. Derfler, Jr.

Bill Blue Moves On

The founder of ABBS starts a message system for Apple owners

In this article, we will visit with one of the major contributors to the microcomputer side of the data communications industry, Bill Blue. We will also look at the Lynx, one new modem system for the TRS-80 and Apple.

Mr. ABBS?

Bill Blue has been referred to in this column and in other places as the father of the Apple Bulletin Board electronic message and program transfer system. There are, by far, more ABBSes operating than any other type of system. But surprisingly, Bill opened this interview by asking to be dropped from the ABBS family tree.

Blue: "It's true that I was heavily involved in the original development of the ABBS software, but I haven't had anything to do with it for over a year. The very first software in skeleton form was written by Craig Vaughn (now with Source Telecomputing Corporation). I took this basic structure and added my own ideas and features. The software was distributed through Peripherals Unlimited, which was Craig's company at the time. He and I worked together on version 3.0, but I haven't had anything to do with the ABBS software since about June 1979."

Microcomputing: "Then what are you doing now?"

Blue: "I have several projects going. All still relate to the Apple computer. My long-term labor of love is the People's Message System. The PMS is totally my concept of what is capable of being done on an Apple system, and I am very proud of the way it has turned out. I originally wrote it just for myself, but recently decided to make it available for others. There are now six PMSes on line and more on the way. Those on line in California include my own in Santee, plus San Diego, Anaheim, Los Angeles and Palo Alto. There is also one on the Gulf Coast at Freeport, TX."

Microcomputing: "What are some of the features of the PMS that make it special? What is the philosophy guiding the features you put in?"

Blue: "I've tried to make the PMS much more than just a message system. I try to be innovative and provide ideas that are new to the field, but still keep the system bullet-proof and easy to operate. This gets tough because the more features you put in the more complex the software becomes.

"For example, I got very tired of having to scan messages and then type in all the message numbers that I wanted to read, so I came up with flagging. The People's Message System was the first to incorporate that feature. All you have to do is type the letter R when you see a message you would like to read later. After you have marked all the messages you want to read, you type an asterisk and they come back in chronological order.

"To get away from the message-only kind of system, I added program downloading—also the first of its kind in the country."

Microcomputing: "Downloading is the ability to transfer programs from the message system to a calling user's Apple?"

Blue: "Yes. And then came program uploading—from the users to the system. I thought it would be a terrific way to exchange programs. Unfortunately, I quickly learned that people were very eager to spend much time in taking programs from the system, but they were very slow to contribute anything. That situation has improved recently, and all the programs I offer on the download section have been contributed by users of the system.

"Lately, on my system in Santee, I have been concentrating more on the "features" section. These range from articles of general interest to editorials contributed by users. The Anaheim PMS is concentrating on the video field, and the Palo Alto PMS is centering around Apple-related articles."

Microcomputing: "That's quite a project. It sounds like you are becoming an electronic publisher. What else has been keeping you busy?"

Blue: "My second big project is the ASCII Express. The Express is a communications program for the Apple that turns it into a smart terminal for data communications operations."

Microcomputing: "We reviewed the program in the July 1980 issue."

Blue: "Yes, and I am still getting orders from that initial review! But now I have a much-updated version I call ASCII Express II. It fully supports upper and lowercase and allows conversational copy as well as a formal file transmission capability. I have included a line editor that can be used to make up messages off line and it has fully programmable keyboard MACROS for very fast semiautomatic log in and command execution."

Microcomputing: "What does it sell for and what kind of equipment do you need to use it?"

Blue: "I thought you'd never ask! It sells for $59.95 and is marketed by Southwestern Data Systems. It is a disk-based program, so you have to have an Apple II with at least one disk drive. You need either Applesoft in ROM or the language card. It works well with the D. C. Hayes modem board, but it can also be

People's Message Systems

Anaheim, CA	714-772-8864	
Freeport, TX	713-223-7943	
Los Angeles, CA	213-291-9314	(Evenings and weekends only)
Palo Alto, CA	415-492-7691	
San Deigo, CA	714-582-9557	(Evenings and weekends only)
Santee, CA	714-449-5689	

META TECHNOLOGIES

FOR YOUR DISK SYSTEM

FILE BOX

DISKETTE STORAGE SYSTEM

$24.95 for 5¼" disks
for 8" disks . . . $29.95

MTC brings you the ULTIMATE diskette storage system, at an affordable price. Storing 50 to 60 diskettes, this durable, smoke-colored acrylic unit provides easy access through the use of index dividers and adjustable tabs. Unique lid design provides dust-free protection and doubles as a carrying handle.

PLASTIC LIBRARY CASES

(not shown)

An economical form of storage for 10 to 15 diskettes, and is suitable for your bookshelf! Case opens into a vertical holder for easy access.

5¼-inch diskette case $3.25
8-inch diskette case $3.50

Single Sided, Single Density, Soft-Sectored 5¼-inch, (for TRS-80™) Mini-floppy

DISKETTES

$21.95 box of 10

These are factory fresh, absolutely first quality (no seconds!) mini-floppies. They are complete with envelopes, labels and write-protect tabs in a shrink-wrapped box.

PLAIN JANE™

DISKETTES
The Beautiful Floppy with the Magnetic Personality™

Thousands of people have switched to this low-cost alternative. These quality diskettes are packaged in a plain white box . . . no fancy printing, fancy names or fancy labels, not even our own (labels cost money). Trust us.

PLAIN JANE™ Diskettes $21.95
10 boxes of 10 . . . . (each box) $21.50

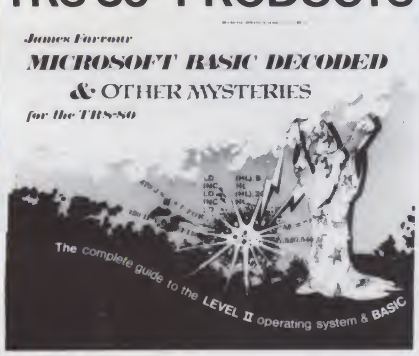
VERBATIM'S PREMIUM DISKETTES AT AFFORDABLE PRICES

DATALIFE™

Seven data-shielding improvements mean greater durability and longer data life. These individually certified diskettes feature thicker oxide coating, longer-lasting lubricant, improved liner, superior polishing and more! Meets or exceeds IBM, Shugart, ANSI, ECMA and ISO standards. Buy the best . . . buy DATALIFE™.

VERBATIM DATALIFE™ DISKETTES
5¼-inch (box of 10)
MD525-01 . . . . $26.95
10 boxes of 10 . . . . (each box) $25.95
8-inch FLOPPIES
Single-Density, FD34-1000 . . . $29.95
Double-Density, FD34-8000 . . $39.95

TRS-80™ PRODUCTS



NEWDOS/80 by Apparat $149.95
NEWDOS + with ALL UTILITIES
35-track $69.95
40-track $79.95
TRS-80™ DISK AND OTHER MYSTERIES $19.95
MICROSOFT ™ BASIC DECODED & OTHER MYSTERIES for the TRS-80™ $29.95

'RINGS' & THINGS

Help prevent data loss and media damage due to improper diskette centering and rotation with the FLOPPY SAVER™ reinforcing hub ring kit. 7-mil mylar rings install in seconds. Kit is complete with centering tool, pressure ring, 25 adhesive backed hub rings and instructions.

HUB RING KIT for 5¼" diskettes . . $9.95
REFILLS (50 Hub Rings) $4.95

Protect your expensive disk drives and your valuable diskettes with our diskette drive head cleaning kit. The kit, consisting of a pair of special "diskettes", cleaning solution and instructions, can be used for 52 cleanings. Removes contamination from recording surfaces in seconds without harming drives.

CLEANING KIT for 5¼" drives . . . $24.95

CALL FOR INFORMATION ON OTHER TRS-80™ PRODUCTS

Products damaged in transit will be exchanged. Prices, Specifications, and Offerings subject to change without notice.

MOST ORDERS SHIPPED WITHIN ONE BUSINESS DAY

PRICES IN EFFECT March 1, 1981 THRU March 31, 1981

WE ACCEPT
- VISA
- MASTER CHARGE
- CHECKS
- MONEY ORDERS
- C.O.D.

- Add $2.50 for standard UPS shipping & handling
- $2.00 EXTRA for C.O.D.
- Ohio residents add 5½% sales tax.

FOR PRODUCT INFO 1-800-321-3640 — **CALL TOLL FREE** — **TO PLACE ORDER 1-800-321-3552**

IN OHIO call (216)289-7500 (COLLECT)

META TECHNOLOGIES CORPORATION

26111 Brush Avenue. Euclid. Ohio 44132

161

810115
TRS-80 is a TM of Tandy Corp.
PLAIN JANE is a TM of MTC.

The Lynx modem by EMTROL Systems provides full-originate and answer capability for the TRS-80 model I.

used with a communications card and standard modem."

Microcomputing: "But that's a finished product. What's next?"

Blue: "Well, the ASCII Express and the People's Message System will never be stagnant. I'm going to keep improving both of them. But for the future, I will soon have a similar terminal package available for the CP/M Apple with the Z-80 softcard. It is called Z-Term and will also be available through Southwestern Data Systems. In addition, I am working on another program I call Online. This is a private message system for those who want to limit access to only authorized users. It will have message upload and download, user account codes, private mail and editing functions."

Microcomputing: "Any words on the outlook for the future of data communications systems?"

Blue: "I'm sure it is obvious to most who have been following your articles or have been participating in this area that we are still just at the starting point of what could be one of the most significant changes in communications since radio and television. This form of communications has been around for years, but only among those "elite few" who were in some way involved with a large company or government computer network. The microcomputer has made this communications form open to everyone. As more persons become computer-literate it will be possible for country-wide communications via computer to be an everyday occurrence in a typical household. I think it's just great and I am thrilled to be a part of it."

(Bill's software products are available from Southwestern Data Systems, PO Box 582, Santee, CA 92071.

The Lynx

I first mentioned the Lynx in the December 1980 Dial-Up Directory (p. 202). The Lynx is a modem for the TRS-80 Level I which does not need the expansion interface to operate. I have used the equipment now for several months and I am impressed with it. At a trade show in December, a Lynx for the Apple II was demonstrated. I don't have any further details right now, but the following description of the TRS-80 Lynx should apply pretty well to the Apple version.

The Lynx plugs directly into the TRS-80 keyboard or the expansion interface. The installation takes about 30 seconds and is well-illustrated in the instruction manual. The Lynx fits nicely under a desk-style telephone and is color-coordinated with the TRS-80.

The modem is well-built, on one high-quality circuit board. The power supply is a typical line cord transformer. Originate and answer modes of operations and the default values of the word length and parity are selected by switches on the back panel. The large Data-Talk switch on the front panel is easy to use, but I would like to see some warning indicator, perhaps an LED or a physical color change, when it is in the data position. Several times family members have tried to use the phone hooked to the Lynx and found it dead because I hadn't flipped back to talk.

The Lynx is a complete system. It comes with its own terminal emulation (EMTERM) software on cassette for either the TRS-80 Level I or II. Even a very barebones Level I system can prepare messages off line, store them for future transmission, and act as a terminal with the ability to send control codes and escape. The Lynx EMTERM program also allows changes to be made to the parity, word length and number of stop bits from the keyboard.

The Store Message command establishes a 1024-character buffer in RAM for writing a message. There is no line editing. Once you get on line with the system or TRS-80 you are exchanging data with, the shift right arrow command will send the message contained in the buffer out the Lynx port. The message buffer is not erased, so the message can be transmitted again to other systems.

The Terminal mode is the normal mode of operation. This dumb terminal emulator provides all the capability you need for operation with all electronic message systems and information utilities. It isn't fancy, but it works! If you have a printer with a parallel interface, it can print received data as it comes in. This is the next best thing to having a disk-based smart terminal saving the data in files. The terminal software includes a command for toggling the printer on and off.

The TRS-80 Level II program adds two more features, which let you transmit and receive BASIC programs. The transmission feature will allow loading a program into RAM from cassette. The program can be transmitted as soon as the receiving station is ready. If you receive a program using the BASIC Receive option, you can then run, list or CSAVE it.

Documentation for the Lynx is like the software—simple but adequate. The eight-page manual tells you what you need to know to operate the system, but not much more. No schematic diagrams or program listings are provided.

The Lynx hardware will also operate with any other software package designed for the standard TRS-80 communications configuration. The Lynx will not work on a TRS-80 that has the standard RS-232 board in place. The inclusion of the practical EMTERM program in the $235 Lynx package makes it a good value. You can easily put a Lynx/TRS-80 Level I terminal on line for under $700. It would be difficult for any commerical supplier to match its performance at that price.

If you have any data communications comments or questions, send them to me at PO Box 691, Herndon, VA 22070. Include a stamped envelope if you want a reply. Electronic mail is welcome at TCB967 on The Source, 70003,455 on CompuServe or the AMRAD CBBS (703-734-1387).□

MICRO QUIZ

What Does This Program Do?

When statement 110 is reached, what is the value of V?

```
 10  DIM X(20)
 20  DATA 6,4,9,2,5,1,8
 30  READ N
 40  FOR I = 1 TO N
 50  READ X(I)
 60  NEXT I
 70  V = X(1)
 80  FOR J = 2 TO N
 90  IF X(J)>V THEN V = X(J)
100  NEXT J
110  PRINT V
120  END
```

(answer on page 150)

Humanitarian Computerists

Over 800 micro users form a network to help refugees

Micros Reuniting Families

An international microcomputer network is helping to reunite Vietnamese families that were separated after emigrating from Indochina.

Family Reunification Services, based in Calgary, Alberta, Canada, includes some 800 microcomputerists in 23 countries. The service, in operation since August 1980, has brought together members of almost 200 families.

The way the service works, each micro user gathers from local humanitarian groups the names of Vietnamese who have resettled in his area and are looking for relatives. He sends the list on tape to the next computerist, who checks it against his list. He adds the names from his area, and mails the cassette to the next station.

If a computerist finds a match, he sends the information to Calgary, and the two parties are notified.

Locating the family members is only the first step. Getting them together is another matter, says FRS President J. E. Caruana, and can be a complicated process.

"Here we get into dollars and cents," Caruana says. "Where do they get the money to travel?

"Also, a person who has resettled is no longer a refugee, and we get into legal and immigration problems. If one family member has to move to another country, does that country have quotas? And what constitutes a family? Is a son through marriage a family member?"

Families sometimes have to relocate over great distances or from one country to another. One family, Caruana says, had to move from Houston, TX, to Montreal, Canada.

Caruana became involved in the refugee movement in 1978. He has since quit his job, and with the financial backing of some local businessmen, runs the FRS full-time. "I work 56 hours a day, 18 days a week, 38 months a year," he says.

The project, he says, will probably go on for another couple of years. "Maybe then we can call the whole thing to a halt and say, 'Hey, we've really accomplished something,' " he says.

What will happen to the network? Caruana, the owner of a TRS-80, hopes to maintain it as a sort of users group that will swap information and ideas on microcomputing in general.

Caruana says that several governments have expressed an interest in the concept of FRS.

"Suppose there were another earthquake in Italy, or an upheaval in Chile," he says. "Something like this could become very useful." But at the moment, Caruana is only interested in reuniting Vietnamese families. "I believe in the cause," he says.

Any microcomputerist, regardless of what kind of system he has, is invited to join the network. For more information, write the FRS, 7203 Huntercrest Road N.W., Calgary, Alberta, Canada T2K 4J9.

Source Offers Job Listings

The Source and Computer Search International are now offering a listing of job opportunities and resumes through The Source's information network.

The CSI "Career Network" is supported by a membership of executive recruiting firms in major cities in the United States, England and Australia. Descriptions of corporate job opportunities and individual resumes are listed. Individuals and corporations can sort through the job market data with reference to salary range, geographic location, job experience, education and special skills. The service includes 40 job categories, displaying the several that apply to any applicant or corporation.

All information listed on the Career Network is confidential. Resumes and job openings are key-coded to the CSI Career Network member recruiting firm that holds the listing. Follow-ups are made directly to the member firm or directly to Computer Search International, in Baltimore.

Peripheral Sales to Soar?

Sales of mini and microcomputer peripherals will increase by 400 percent over the next ten years, claims a recent report from International Resource Development, Inc.

In addition, the IRD report forecasts that:

- Floppy disk drives, Winchester drives and other mass storage devices will remain the single largest segment of the market.

Peripheral Category	1980	1982	1985	1990
Data Entry Equipment	240	370	915	1,730
Printers	560	780	1,010	1,185
Memory	300	750	1,150	1,650
Disk/Tape	1,000	1,850	2,565	2,670
G P Terminals	650	1,100	1,650	3,450
Datacomm Equipment	370	665	925	1,915
Sub-total	3,120	5,515	8,215	12,600
Integral Peripherals	240	260	2,115	17,400
Totals	3,360	5,775	10,330	30,000

Mini/micro peripheral shipments for 1980-1990. (Source: International Resource Development, Inc.)

- Total printer shipments for mini/micro systems will jump from $500 million in 1980 to $1 billion in 1990. Half of those printers will be nonimpact.
- More and more microcomputer products will have built-in telecommunications abilities and will be used to access a variety of new information services and data bases.
- While there is now about one terminal for each 20 white collar desks, the proportion will be one in three by the end of the decade.
- More than half of the 100+ firms now making mini/micro peripherals will disappear.
- Xerox, which owns Shugart and Diablo, could become the leading supplier of mini/micro peripherals in the 80s.

For more information on the report, contact IRD at 30 High St., Norwalk, CT 06851 (Tel. 203-866-6914, Telex 64 3452).

School Pushes Telecom

Golden Gate University in San Francisco is going into telecommunications in a big way. The school has established two new undergraduate programs in telecommunications management and has appointed four telecommunications professionals to the faculty.

Students will be offered a B.S. in telecommunications management, as well as graduate and undergraduate certification in telecommunications management if they don't want the degree.

Courses include Introduction to Telecommunications Management, Corporate Telecommunications, Survey of Data Communications Systems, Networks and Switching Systems, Legal and Regulatory Aspects of Telecommunications, Telecommunications Technology and Society, Teleconferencing and Satellite Systems, and Viewdata and On-line Commercial Information Systems.

For information, write GGU, 536 Mission St., San Francisco, CA 94105.

TV Station Uses Computer for Elections

WOWK-TV in Huntington, WV, used a small business computer to tabulate election returns during the Presidential elections in November.

Returns were phoned in to WOWK by League of Women volunteers from precincts in the West Virginia-Ohio-Kentucky triangle. Station personnel then keyed the figures into a Pertec PCC 2000 system. Four terminals computed the returns, while a fifth broadcast the results to home viewers.

The old method is to tabulate returns manually. Incoming results are flashed in handwritten numbers, which are then entered into the video equipment. The PCC 2000 provided more accuracy, and also let viewers watch the results change on the terminal.

Vroom Vroom

The computerized motorcycle has arrived.

The Yamaha SECA 750 uses a microcomputer to monitor the sidestand, the brake fluid level, the battery fluid level, the fuel level, the engine oil level, the headlight, the taillight and the stoplight.

The system consists of the micro, an integrated circuit board, a liquid crystal diode and sensors for the key functional areas being monitored.

When the driver turns on the ignition, the handlebar-mounted computer first reports its condition before beginning to analyze the sensored areas two seconds later. When the monitored function is in working order, there is a brief burst of light on the LCD confirming this. But, if a problem—however minor—arises in any of the functions subject to the computer's scrutiny, the LCD remains lit.

Suggested retail price is expected to be less than $3500.

Videotex '81

Videotex '81, the first major conference and exhibition on videotext in North America, is set for May 20–22 in Toronto, Ontario.

The conference program will run in three parallel streams. The first will cover current videotext developments around the world, and will include reports from the UK, France, Germany, Japan, Canada and the US. The other two will be for executives looking for new and more effective corporate information systems and for businessmen looking for opportunities as entrepreneurs, information providers and equipment suppliers.

Conference topics will include advertising through videotext, teleshopping and direct marketing, electronic mail, electronic funds transfer, education, entertainment, database publishing, impact on newspapers, private videotext systems, syndicated user groups, interactive cable TV, teletext, intelligent terminals and personal computers, regulatory and legal issues, market development projections and the social impact of new technology.

For more information, contact Pam Carter, Videotex '81, Infomart, 122 St. Patrick St., Toronto, Ontario M5T 2X8, Canada (Tel. 416-598-1981, Telex 0622111).

Quote of the Month

"We might stop having printed magazines, and everything might appear on a screen someplace, but I don't think that that's any less a magazine. If people prefer to read it sitting in front of a screen, fine, we'll create such an editorial product."

—Edward Thompson, Editor in Chief of *Reader's Digest*, which recently bought The Source information network; quoted in *Next* magazine.

The Yamaha SECA 750, the world's first computerized motorcycle. The computer console is shown in the inset.

AS THE WORD TURNS By Eric Maloney

Etymological Gymnastics

Several years ago, the newspapers covered a number of dramatic stories about parents hiring kidnappers to rescue their children from unorthodox religious groups. These kidnappers would take the children into a motel room for several days and nights in an attempt to "deprogram" them.

Nobody gave the word "deprogram" much thought at the time. But it is a prime example of how the computer world can borrow a word, change the meaning and reintroduce it into everyday language.

The original meaning of the verb "program" is, according to the *Oxford English Dictionary*, "To arrange by or according to a program; to draw up a program of; to scheme or plan definitely." They illustrate with an example from the sports world: "This match was programmed to start yesterday."

When computerists took the word for their own use, they modified and narrowed its definition. "Program" came to mean the act of preparing a set of instructions telling a computer what to do. A program, says Charles J. Sippl's *Microcomputer Dictionary and Guide*, includes "plans for the transcription of data, coding for the computer and plans for the absorption of the results into the program."

Eventually, computers caught the attention of the general public, and people became familiar with the concept of programming. It was only a matter of time before the idea was applied to humans, as a synonym for indoctrination or brainwashing.

(It is interesting that programming, in itself a neutral act, has come to indicate a cold, inhuman, ultimately destructive process. This says something about how people perceive computers. I'll deal with this in a later column.)

Several other computer words are working their way into the common language. "Interface" is an example.

The *OED*, published in 1935, defines an interface as "a surface lying between two portions of matter or space." Scientists extended that meaning to include, according to the 1976 *OED* supplement, "An apparatus designed to connect two scientific instruments, devices, etc., so that they can be operated jointly."

Finally, the social scientists applied it to the field of human experience, using the word to indicate "A means or place of interaction between two systems, organizations, etc.; a meeting-point or common ground between two parties, systems or disciplines; also, interaction, liaison, dialogue."

"Interface" has since become a buzzword meaning "to talk." It is interchangeable with "rap," "relate" or "interact." For example: "Why don't we go to lunch, and interface?"

"Peripheral" is another one. Originally an adjective, it means "Of, pertaining to, or situated in, the periphery . . . . of the surface or outward part of an organic body, especially in peripheral neuritis, inflammation of one or more nerves of both sides." Computerists made it a noun, meaning any device that works with a computer but is not a part of it.

This word isn't used as much as "program" or "interface," but it's only a matter of time. I predict that within ten years, we'll be buying peripherals for our blenders, outdoor barbecue sets, trash compacters, and probably even our wristwatches.

What's next? "Modem" has a good chance of making it into our everyday language, as soon as enough businessmen start using them. Other possibilities include "debug," "buffer" and "software."

Then there are all the possible new idiomatic expressions. Someone not in the mainstream of office activities is "out of the loop." A person no longer in complete control of his senses is characterized as having "blown a disk."

If you've heard any interesting new expressions using computer terminology, drop me a line. I'll put them in my buffer for a future column.□

The Newest In

Apple Fun

We've taken five of our most popular programs and combined them into one tremendous package full of fun and excitement. This disk-based package now offers you these great games:

Mimic—How good is your memory? Here's a chance to find out! Your Apple will display a sequence of figures on a 3×3 grid. You must respond with the exact same sequence, within the time limit.

There are five different, increasingly difficult versions of the game, including one that will keep going indefinitely. Mimic is exciting, fast paced and challenging—fun for all!

Air Flight Simulation—Your mission: Take off and land your aircraft without crashing. You're flying blind —on instruments only.

A full tank of fuel gives you a maximum range of about 50 miles. The computer will constantly display updates of your air speed, compass heading and altitude. Your most important instrument is the Angle of Ascent/Bank Indicator. It tells if the plane is climbing or descending, whether banking into a right or left turn.

After you've acquired a few hours of flying time, you can try flying a course against a map or doing aerobatic maneuvers. Get a little more flight time under your belt, the sky's the limit.

Colormaster—Test your powers of deduction as you try to guess the secret color code in this Mastermind-type game. There are two levels of difficulty, and three options of play to vary your games. Not only can you guess the computer's color code, but it will guess yours! It can also serve as referee in a game between two human opponents. Can you make and break the color code . . .?

Star Ship Attack—Your mission is to protect our orbiting food station satellites from destruction by an enemy star ship. You must capture, destroy or drive off the attacking ship. If you fail, our planet is doomed . . .

Trilogy—This contest has its origins in the simple game of tic-tac-toe. The object of the game is to place three of your colors, in a row, into the delta-like, multi-level display. The rows may be horizontal, vertical, diagonal and wrapped around, through the "third dimension". Your Apple will be trying to do the same. You can even have your Apple play against itself!

Minimum system requirements are an Apple II or Apple II Plus computer with 32K of memory and one minidisk drive. Mimic requires Applesoft in ROM, all others run in RAM or ROM Applesoft.

Order No. 0161AD $19.95

Solar Energy For The Home

With the price of fossil fuels rising astronomically, solar space-heating systems are starting to become very attractive. But is solar heat cost-effective for you? This program can answer that question.

Just input this data for your home: location, size, interior details and amount of window space. It will then calculate your current heat loss and the amount of gain from any south facing windows. Then, enter the data for the contemplated solar heating installation. The program will compute the NET heating gain, the cost of conventional fuels vs. solar heat, and the calculated payback period—showing if the investment will save you money.

Solar Energy for the Home: It's a natural for architects, designers, contractors, homeowners. . .anyone who wants to tap the limitless energy of our sun.

Minimum system requirements are an Apple II or Apple II Plus with one disk drive and 28K of RAM. Includes AppleDOS 3.2.

Order No. 0235AD (disk-based version) $34.95

Math Fun

The Math Fun package uses the techniques of immediate feedback and positive reinforcement so that students can improve their math skills while playing these games:

Hanging—A little man is walking up the steps to the hangman's noose. But YOU can save him by answering the decimal math problems posed by the computer. Correct answers will move the man down the steps and cheat the hangman.

Spellbinder—You are a magician battling a computerized wizard. In order to cast death clouds, fireballs and other magic spells on him, you must correctly answer problems involving fractions.

Whole Space—Pilot your space craft to attack the enemy planet. Each time you give a correct answer to the whole number problems, you can move your ship or fire. But for every wrong answer, the enemy gets a chance to fire at you.

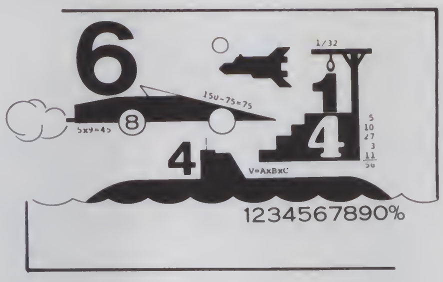
Car Jump—Make your stunt car jump the ramps. Each correct answer will increase the number of buses your car must jump over. These problems involve calculating the areas of different geometric figures.

Robot Duel—Fire your laser at the computer's robot. If you give the correct answer to problems on calculating volumes, your robot can shoot at his opponent. If you give the wrong answer, your shield power will be depleted and the computer's robot can shoot at yours.

Sub Attack—Practice using percentages as you maneuver your sub into the harbor. A correct answer lets you move your sub and fire at the enemy fleet.

All of these programs run in Applesoft BASIC, except Whole Space, which requires Integer BASIC.

Order No. 0160AD $19.95



Skybombers

Two nations, seperated by The Big Green Mountain, are in mortal combat! Because of the terrain, their's is an aerial war—a war of SKYBOMBERS!

In this two-player game, you and your opponent command opposing fleets of fighter-bombers armed with bombs and missiles. Your orders? Fly over the mountain and bomb the enemy blockhouse into dust!

Flying a bombing mission over that innocent looking mountain is no milk run. The opposition's aircraft can fire missiles at you or you may even be destroyed by the bombs as they drop. Desperate pilots may even ram your plane or plunge into your blockhouse, suicidally.

Flight personnel are sometimes forced to parachute from badly damaged aircraft. As they float helplessly to earth, they become targets for enemy missiles.

The greater the damage you deal to your enemy, the higher your score, which is constantly updated at the bottom of the display screen.

The sounds of battle, from exploding bombs to the pathetic screams from wounded parachutists, remind each micro-commander of his bounden duty. Press On, SKYBOMBERS—Press On!

Minimum system requirements: An Apple II or Apple II Plus, with 32K RAM, one disk drive and game paddles.

Order No. 0271AD (disk-based version) $19.95

Paddle Fun

This new Apple disk package requires a steady eye and a quick hand at the game paddles! It includes:

Invaders—You must destroy an invading fleet of 55 flying saucers while dodging the carpet of bombs they drop. Your bomb shelters will help you—for a while. Our version of a well known arcade game! Requires Applesoft in ROM.

Howitzer—This is a one or two person game in which you must fire upon another howitzer position. This program is written in HIGH-RESOLUTION graphics using different terrain and wind conditions each round to make this a demanding game. The difficulty level can be altered to suit the ability of the players. Requires Applesoft in ROM.

Space Wars—This program has three parts: (1) Two flying saucers meet in laser combat—for two players, (2) two saucers compete to see which can shoot out the most stars—for two players, and (3) one saucer shoots the stars in order to get a higher rank—for one player only. Requires Applesoft.

Golf—Whether you win or lose, you're bound to have fun on our 18 hole Apple golf course. Choose your club and your direction and hope to avoid the sandtraps. Losing too many strokes in the water hazards? You can always increase your handicap. Get off the tee and onto the green with Apple Golf. Requires Applesoft.

The minimum system requirement for this package is an Apple II or Apple II Plus computer with 32K of memory and one minidisk drive.

Order No. 0163AD $19.95

Instant Software™

*A trademark of Apple Computer Inc.

PETERBOROUGH, N.H. 03458
603-924-7296

Apple* Software

From Instant Software

Santa Paravia and Fiumaccio

Buon giorno, signore!

Welcome to the province of Santa Paravia. As your steward, I hope you will enjoy your reign here. I feel sure that you will find it, shall we say, profitable.

Perhaps I should acquaint you with our little domain. It is not a wealthy area, signore, but riches and glory are possible for one who is aware of political realities. These realities include your serfs. They constantly request more food from your grain reserves, grain that could be sold instead for gold florins. And should your justice become a trifle harsh, they will flee to other lands.

Yet another concern is the weather. If it is good, so is the harvest. But the rats may eat much of our surplus and we have had years of drought when famine threatened our population.

Certainly, the administration of a growing city-state will require tax revenues. And where better to gather such funds than the local marketplaces and mills? You may find it necessary to increase custom duties or tax the incomes of the merchants and nobles. Whatever you do, there will be far-reaching consequences . . . and, perhaps, an elevation of your noble title.

Your standing will surely be enhanced by building a new palace or a magnificent *cattedrale*. You will do well to increase your landholdings, if you also equip a few units of soldiers. There is, alas, no small need for soldiery here, for the unscrupulous Baron Peppone may invade you at any time.

To measure your progress, the official cartographer will draw you a *mappa*. From it, you can see how much land you hold, how much of it is under the plow and how adequate your defenses are. We are unique in that here, the map IS the territory.

I trust that I have been of help, signore. I look forward to the day when I may address you as His Royal Highness, King of Santa Paravia. *Buona fortuna* or, as you say, "Good luck". For the Apple 48K.

Order No. 0174A $9.95 (cassette version).
Order No. 0229AD $19.95 (disk version).

TO ORDER

SEE YOUR LOCAL INSTANT SOFTWARE DEALER OR USE THE ORDER FORM BELOW

For Fast Service *call now* Toll-Free 1-800-258-5473

Apple Cassettes

0018A Golf $7.95
0025A Mimic $7.95
0040A Bowling/Trilogy $7.95
0073A Math Tutor I $7.95
0079A Oil Tycoon $9.95
0080A Sahara Warriors $7.95
0088A Accounting Assistant $7.95
0094A Mortgage w/Prepayment Option/ Financier $7.95
0096A Space Wars $7.95
0098A Math Tutor II $7.95
0174A Santa Paravia and Fiumaccio $9.95
0148A Air Flight Simulation $9.95

We Guarantee It!

Instant Software Guarantee

OUR PROGRAMS ARE GUARANTEED TO BE QUALITY PRODUCTS. IF NOT COMPLETELY SATISFIED YOU MAY RETURN THE PROGRAM WITHIN 60 DAYS. A CREDIT OR REPLACEMENT WILL BE WILLINGLY GIVEN FOR ANY REASON.

Name ______________________

Address ______________________

City ______________ State ________ Zip ________

☐ Check ☐ Money Order ☐ VISA ☐ AMEX ☐ Master Charge

Card No ______________ Exp. Date ________

Signed ______________ Date ________

Order your Instant Software today!

Quantity	Order No	Program name	Unit cost	Total cost
		Shipping and handling		$1.00
			Total order	

Instant Software™ Inc.
Peterborough, N.H. 03458

✓40

Computer, terminal and mass-storage device—Heathkit's All-In-One. First of two parts.

Building the H-89

By Martin Moore

Perhaps you're in the situation where you want a computer, you need a terminal and you need mass storage. But you don't want to pay three separate prices for three separate boxes—that may or may not work together.

Well, that was my predicament. I wanted a system for text processing. I wanted a fairly fast computer, a high-quality terminal with a decent typewriter-style keyboard and a floppy disk. Most of all, I wanted it in one box that wouldn't take up all my desk space. And I wanted it for under $1500.

I had about given up when in late May of 1979 our friendly postman delivered into my hands a copy of the Heath catalog. And right in the middle was just what I wanted. Heath billed it as the H-89 "all-in-one" computer—a Z-80 controlled terminal, a Z-80-based computer and a 5-1/4 inch floppy drive. All in a small package. Terrific! I ordered one in mid-June of that year. It all came together in mid-October.

Building the Heathkit H-89

What arrived on my doorstep weighed 86 pounds and was distributed among three boxes. The first thing I noticed was that one of the boxes had arrived with its sides bashed in. With some trepidation, I gingerly opened the package to find a 12-inch CRT. Incredibly, the box had been crushed in around the neck of the tube, leaving the glass unscathed.

The smallest box contained a Wangco Model 82 flexible disk drive. The largest box contained the H-89 itself. Inside the large box were two smaller boxes and the computer chassis. One of the smaller boxes, the "open me first" box, contained the assembly and operating manuals. The first rule of kit building is to read the assembly manual all the way through—before you begin to build the kit. This little step saves you a lot of headaches. If you want it to work the first time, read the book.

The assembly manual was laid out in Heath's usual excellent fashion. After adding the errata sheets (no one's perfect), I went to work.

Each step of construction is clearly explained, giving detailed information about each phase of assembly. If you've had to contend with other manufacturers' assembly instructions, you'll appreciate Heath. After checking to see that all the parts were there, all I needed was a large work space devoid of small children.

A Foam Cabinet?

The first task is to mount the power transformer on the cabinet base. And yes, Virginia, the cabinet is foam (see Photo 1). Heath chose to save weight by using a structural foam cabinet. Structural foam is curious stuff. It has a hard appearance, but is fairly flexible. It's lightweight, sturdy and easy to work with. You'd be hard-pressed to break it. But bang it with a sharp object, and you'll gouge out material.

Before mounting hardware to the base, brass inserts must be implemented in holes provided. The inserts are placed part way into the holes, then heated to melt their way into the foam as shown in Photo 2. My initial concerns about strength were alleviated when I found I could tighten screws to my heart's content, without stripping the insert out of the foam.

The cabinet top mounts to the base by way of two hinges at the rear. The hinges allow the cabinet top to be raised or removed entirely for circuit access.

Circuit Boards

When you order the H-89, you get:

- a power supply board,
- a video driver board,
- a Z-80-based terminal logic board (prewired),
- a Z-80 CPU board (prewired),
- a cassette tape interface board (prewired),
- a floppy disk controller board (prewired) and
- the floppy disk drive.

So only two circuit boards had to be built—two circuit boards and what seemed like a whole raft of connector cables. The biggest challenge with this kit was trying to hold #22 wire in a connector and solder it at the same time. I ended up with blisters on my fingertips. I suggest that anyone thinking about buying an H-89 invest in a small vice to hold the connector body while you solder.

The Power Supply

The power supply board is straightforward in construction, requiring

Martin Moore (2735 SW 229, Aloha, OR 97006) contributed the article "The 16-Bit Super Processors Are Here" to our August 1980 issue.

only eight diodes, four filter capacitors and three pin connectors. Make sure you double-check the polarity on the diodes and capacitors. The smell of burning caps is so depressing.

The power transistors mount on a large aluminum heat sink that stands alongside the power supply board. For additional cooling, the fan is mounted on the cabinet top right over the power supply.

Once the supply is built, static tests (power off) are performed to see if any large globs of solder were left on the board. If your job passes the static tests, the moment of truth arrives. You plug it in. Fortunately, mine worked the first time.

The power supply provides +65 V to the video driver board, and +18, −18, +5 and +8.5 V to the rest of the boards. The Wangco disk drive gets +5 and +12 V.

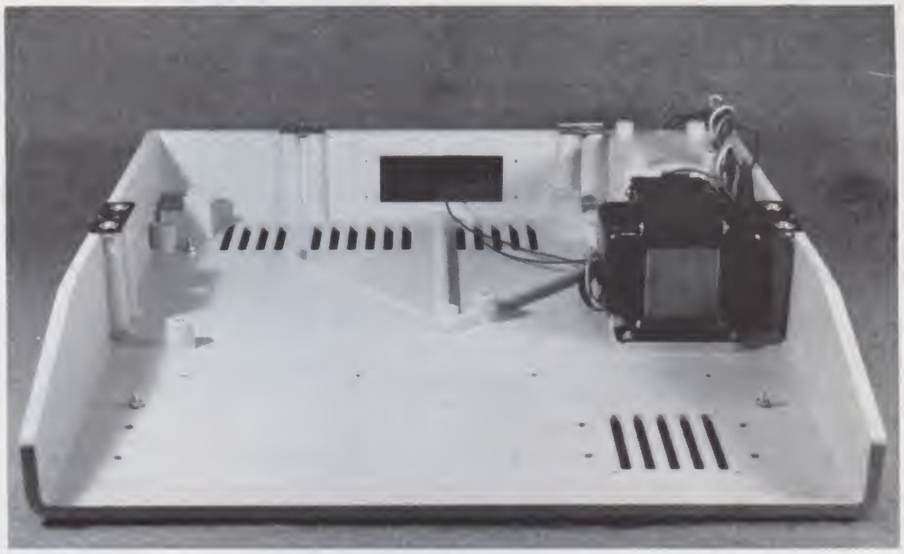


Photo 1. The structural foam base with power supply transformer mounted. The entire H-89 cabinet is made of structural foam.

Video Driver Board

The video driver board is next. This board requires about three hours of construction time. Once again, if you follow Heath's instructions, you'll have no trouble. All the part locations are clearly marked on the board, so it's difficult to get a component in the wrong spot.

I did have one bit of trouble. The circuit uses quite a few small glass-encapsulated diodes. Theoretically, the manual gives you the part number, and you examine the diodes for that number. They even give you a magnifying glass so you can see the number. In actual fact, it's nearly impossible to see the number printed on the diode. It seemed like I spent most of my time trying to make out what kind of diode I had. If Heath is open to advice, I'd suggest that they wrap all like diodes together and mark the package. That would certainly save on eye strain. As it was, all the components were in one plastic sack, sort of jumbled together.

Circuitry—The video driver board has its own voltage regulator, which produces a stable +53 and +6 V. The +53 V supply is used to elevate the CRT cathode. The video signals from the terminal logic board, after passing through a cascode amplifier, ride on this +53 V level.

The vertical sync square wave from the terminal logic board is shaped to create a sweep ramp signal that is amplified and placed on the vertical yoke of the CRT. A vertical size adjustment in the amplifier will let you set the vertical height of the display to whatever is comfortable. For me, comfortable was about an inch shorter than the six inches recommended by Heath.

The horizontal sync pulse from the terminal logic board is used to create the +500 V anode voltage for the CRT, as well as control the horizontal deflection of the electron beam.

Placement—The video driver board rests flat in the bottom of the cabinet (Photo 3), mounted to the base by four screws. There's only one problem with this placement, and that occurs later, when you have to slide the cable from the keyboard under the video driver board. It's somewhat difficult to convince the little beggar that it ought to go under there.

Terminal Logic Board

The terminal logic board did not require any work on my part, other than to prepare some more of those countless cables. The terminal logic board is a computer in itself. It has a Z-80 microprocessor, Motorola's 6845 CRT controller, an asynchronous serial port, program ROM, scratchpad RAM and display memory. I'll talk more about the abilities of the terminal later on.

I had one problem with the terminal logic board, and at the time I thought it was my error. The board generates three signals called video, vertical sync and horizontal sync. These three signals are fed to the video driver board (the one I put together).

When I first powered up the H-89, the raster looked horrible. Instead of white letters on a black background, I was getting white letters on a splotchy gray background. And I could see the retrace zipping across the screen. The problem was fairly obvious—there wasn't enough video signal to create decent contrast. If I backed the brightness down, the characters disappeared. Turn the brightness up, and the screen turned gray.

The oscilloscope told the story. The video driver board was getting nice, fat 5 V signals from the vertical sync and horizontal sync lines, but the video signal was almost nonexistent. Back to the terminal logic board.

Heath uses 74LS86 exclusive-OR devices to drive the video signals. And one of the gates had died. Once I replaced the 74LS86, the display was beautiful—one of the sharpest displays I've seen. Actually, it's a wonder there was any display at all. The

Photo 2. Threaded brass inserts are melted into the foam cabinet. The inserts are used to mount hardware to the cabinet.

Photo 5. The H-89 graphic characters.

keypad lets you do on-screen editing (although none of the Heath software supports this function).* Keys 1 through 9 each have a shifted function that controls the cursor and performs character and line editing jobs. Shifts 4, 8, 6 and 2 move the cursor left, up, right and down, respectively. Shift 3 will delete the cursor line. Shift 7 will let you insert characters, moving all the characters to the right of the cursor to the right of the screen. Shift 9 causes the character at the cursor position to be deleted. And shift 5 homes the cursor to the upper left of the screen.

If you don't want to press the shift key each time you use the cursor controls, the terminal can be placed into a shifted keypad mode by entering ESC t (escape t). This can be done either off-line at the keyboard, or by the computer. This is one of the many escape sequences used to program the keyboard and terminal. ESC t places the keypad into the shifted mode until you tell it otherwise, or shut the H-89 off. Or, if you think you'll want the keypad in the shifted mode most of the time, there's a switch on the terminal logic board that will cause the terminal to power up in the keypad shifted mode. Then you can use ESC u to get out of the shifted mode.

There's a third mode, called the alternate keypad mode, into which the keypad can be programmed. This mode causes the keypad to act much like the programmable keys mentioned before. In the alternate keypad mode, each key sends an unused escape sequence to the computer.

None of these codes are supported in Heath software, but instead rely on a program you write to perform a specific function. This is not to say that Benton Harbor BASIC, Microsoft BASIC or Heath CP/M won't work with these keys. It's just that you have to tell your BASIC program to look out for them. All in all, you get 19 programmable keys with this terminal.

Escape sequences—I mentioned escape sequences in regard to the eight programmable keys and the keypad. Actually, the whole terminal is controlled with escape sequences, either sent from the computer or entered on the keyboard. The escape sequences do the following:

- ESC p puts the terminal in reverse video mode
- ESC q gets the terminal back out of reverse video
- ESC F puts the terminal in the graphics mode
- ESC G exits the graphics mode
- ESC t places the keypad into its shifted mode
- ESC u exits the shifted mode
- ESC = places the keypad into its alternate mode
- ESC > gets the keypad back to its normal mode
- ESC [puts the terminal into the hold screen mode
- ESC \ exits the hold screen mode

All of the cursor control key actions can be duplicated by the computer with escape sequences. There's an escape sequence for each cursor movement (up, down, backward, forward, home). If the computer sends the terminal an ESC n, the terminal tells the computer where the cursor is with an x-y coordinate. ESC j causes the terminal to save the present location of the cursor, which can be returned to the saved location with an ESC k. And the cursor can be moved directly to a position by entering ESC Y, followed by an x-y coordinate (direct cursor addressing).

All of the editing functions (erase page, erase part of the page, erase a line, insert a line, etc.) can be generated by the computer with an escape sequence.

Escape sequences also let the computer reconfigure the terminal. For example, ESC r lets you set the RS-232C data rate from 110 to 9600 baud. ESC x lets you enable the 25th line, disable the key click, change the cursor from an underline to a block, shut the cursor off and go into any of the shifted or alternate modes. ESC y lets you change everything you set with ESC x. And ESC z resets the whole works to its power-up configuration. It's a very busy terminal!

The one irritating aspect of this keyboard is that a repeat key is used instead of implementing automatic repeat on each key. If you want multiples of the same character, you have to hold down the repeat key in addition to the character key.

The Screen

The H-89 screen in my machine is of Toshiba origin, and will display 25 lines of 80 characters with true descenders. The screen definition is sharp—you can see each dot in the display's matrix. In fact, trying to blur the screen with the focus control is nearly impossible. I turned the focus control to both stops, and could see very little change in the display. This is a high-quality CRT.

Graphics

The H-89 cannot handle true graphics. Instead, what you get are 33 graphic symbols (see Photo 5). The 33 symbols available are flexible enough to do moderately sophisticated displays, but you simply can't do true, point-to-point graphics work. What's missing is the software in the terminal. I believe the hardware is there, and the screen certainly has enough resolution to handle 2000 point graphics.

But don't despair. I'm sure some enterprising software expert will come up with a ROM package that will let the terminal do its stuff.

The CPU Half

The computer portion of the H-89

Photo 6. The CRT and facade are mounted. The ribbon cable connects to the terminal keyboard.

Photo 7. Nearly complete, the H-89 only requires the 5.25-inch disk drive to be mounted.

is, in all respects, a general-purpose computer. A 2 MHz Z-80 runs the whole affair. The H-89 comes standard with 16K of RAM based at 040,100 (split octal notation). Below this memory space resides the system ROM and the floppy disk ROM and RAM. Heath has reserved 3K of the low memory address space for future use.*

The hardware is already on the board, waiting for another 32K of RAM, which will let you expand to 48K.

There are several I/O ports on the CPU logic board. One port, called the general-purpose port, is for internal use. The general-purpose port tells the computer the state of the configuration DIP switch (baud rate settings, etc.). Another port on the CPU logic board traps on all H8 software instructions that would normally go to the H8 front panel (you can directly run Heath's H8 software on the H-89 without modifications). Since the H-89 doesn't have a front panel, these port accesses are redirected to either do nothing or cause a display on the terminal.

A console port handles serial data transfer to the terminal half of the H-89. And, like the terminal half, the computer uses an 8250 ACE for communications.

The cassette port is actually the place where the cassette interface board plugs in. The cassette port is controlled by firmware in the H-89 system ROM.

The H-89 CPU logic board has room for six accessory circuit boards. Each board plugs into square pins on the CPU logic board. The square pins make up the six parallel I/O ports.

Cassette Tape Interface

When you order the H-89, you get the cassette tape interface with it, whether you want it or not. This seems a little redundant in the face of the floppy disk controller and drive, but that's the way it is.

The cassette interface board converts digital information into a Kansas City-type 1200 Hz/2400 Hz format. The information is recorded and loaded at 1200 baud. The interface can drive two recorders, reading information from one and storing information on the other. Or, you can use just one recorder. When you're loading information onto a tape, the interface board automatically puts a three-second tone on the tape before data transfer begins. The tape operating system looks for this three-second tone when the data is loaded back into the computer.

Another feature of the cassette interface board is that you can reduce the incoming signal by 500 by moving a jumper on the board. The incoming signal is normally about 500 mV peak-to-peak. Installing the jumper reduces the signal to about 1 mV peak-to-peak.

The cassette interface board

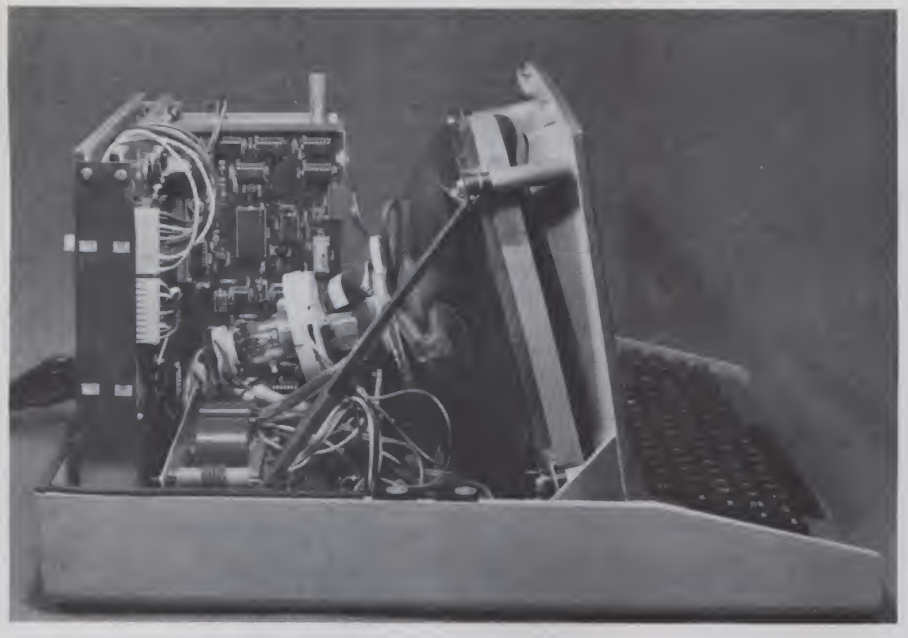


Photo 8. A side view of the H-89. The CPU logic board and terminal logic board run across the rear of the unit. The small board is the cassette interface board. The disk drive is not yet mounted.

mounts onto the CPU logic board. The interface board comes assembled, but you have to put together the cabling from the board to the rear panel.

The LED curiosity—When I mounted the cassette interface board, I noticed an LED lurking around the middle of the board. I looked in the operator's manual to find out about the LED, but was disappointed. It said nothing. The theory of operation was silent on the matter, too.

Well, it turns out that the LED is connected to the input wave shaping circuitry, and indicates whether the information is coming through clearly. If you watch the LED when you're loading information from cassette, you can tell whether the incoming signal is too strong or too weak. If the signal is too strong, the LED glows steadily. If the signal is too weak, the LED doesn't come on at all. A flickering light is just about right. This lets you adjust the volume control on your recorder to the optimum setting.

The Disk Drive and Controller

The drive used in the H-89 is the Model 82 from what used to be the Wangco Division of Perkin-Elmer, before Perkin-Elmer sold the line to Seimens. The drive uses a ten-hole hard-sectored disk, and can store onto 40 tracks. This is a pretty good drive. If you can come up with a controller that will use double-density encoding methods, you can record double-density. Heath's controller uses single-density recording.

Another nice thing about the drive is that it can be daisy-chained to up to three other drives. That is, you can connect three other drives to the original, without altering the controller board. Heath software currently supports a maximum of three drives. Heath will sell you another two drives with enclosure and power supply for $895.

The drive is turned off and on by software. This means that you don't have to listen to the drive rotate when it's not doing anything. There is a one-second delay, however, from the time the drive is turned on until it's up to operating speed.

The controller—Heath chose to use an S2350 USRT as the basic controlling element in their disk controller. The S2350 converts the bus parallel data to nonreturn to zero (NRZ) serial data. Other circuitry on the controller converts the NRZ data to the bit cell format required by the drive. The beauty of this controller is its simplicity. No fancy gimmicks or bells and whistles. In other words, not much to go wrong.

Drive calibration—Because the drive is a mechanical device, it will wear with age. One of the things that wears is the drive motor. Rotational speed of the motor must be constant, and must be within set limits. Heath foresaw the need to be able to calibrate the drive motor periodically, and put a handy little routine in their firmware that calculates the motor's rotational speed. To adjust the drive speed, you pull the drive part way out of the cabinet and turn a potentiometer on the drive, while watching numbers on the H-89 screen.

Firmware/Software —The H-89 Monitor

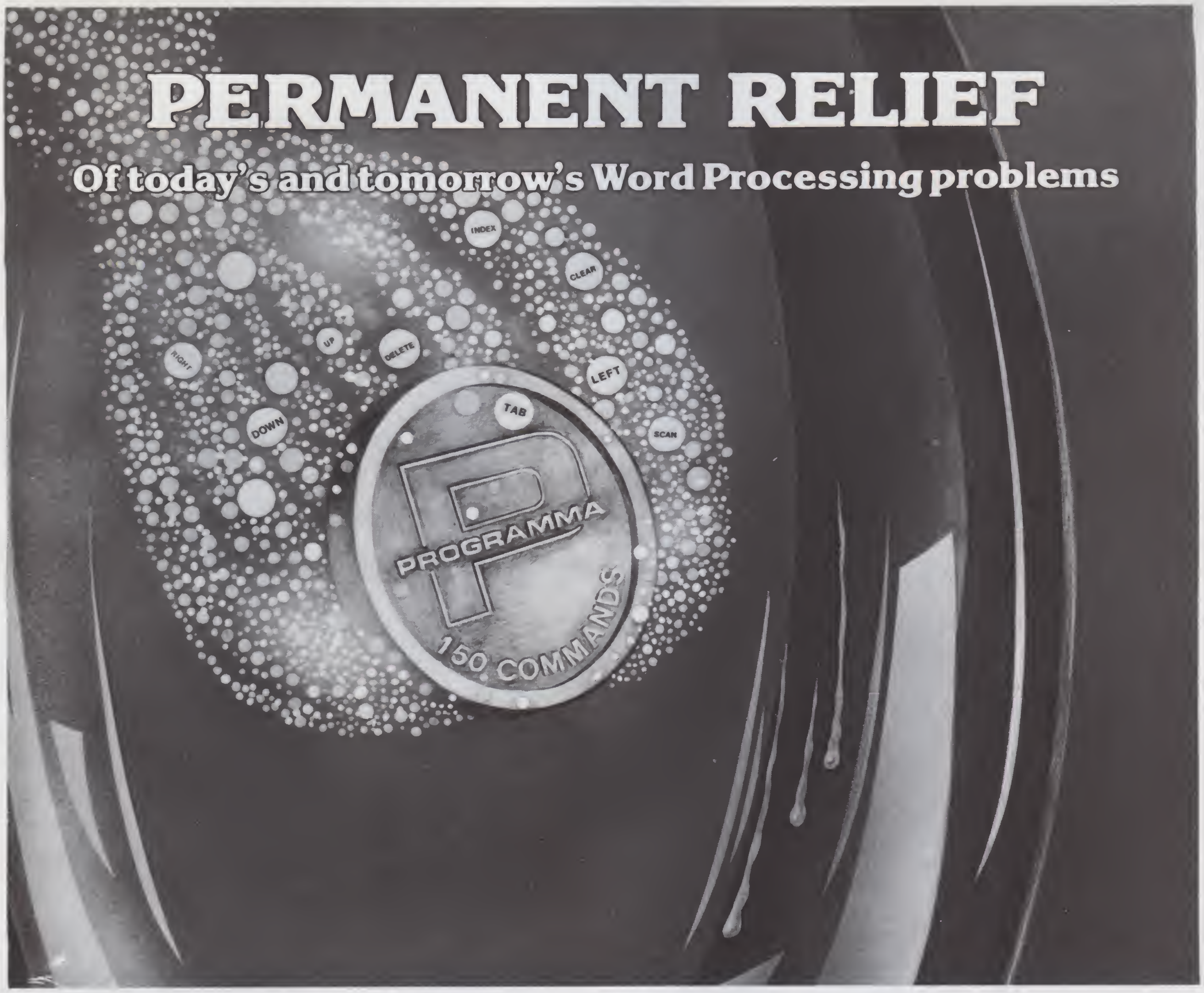
The H-89 monitor exists in ROM at the bottom 2K of address space.* (This, by the way, is one of the major complaints about the H-89 and H8. A lot of other Z-80 or 8080 computers use software that requires the bottom of memory to be RAM. Heath's monitor precludes easy conversion of other people's software.) The monitor is somewhat primitive, but lets you communicate with the computer without loading external software. You can even enter and execute programs, examine and alter memory, and store and retrieve programs from tape.

When the power is turned on, the H-89 beeps twice, then shows H: on the terminal. The H: is the prompt character. All command entries to the monitor are single-character. The H-89 fills in the rest of the command. For example, if you press "S" (no return required), the terminal will print "SUBSTITUTE." Then it's up to you to enter the address you want to look at or change. The following is a list of monitor commands that can be entered from the keyboard.

Substitute—This command will let you examine or change a value in RAM. You press "S" and enter the address in split-octal notation. As soon as you've entered the address and pressed return, the monitor places the address and its contents on the terminal screen. At this point, you can change the memory contents, or you can leave them the way they are.

You can look at and change preceding or successive addresses by pushing the dash key or the space bar, respectively. You can quickly cycle through several addresses by holding down the space bar or dash key, and pressing the repeat key. If you want to change a memory value, simply enter the value. If you make a mistake with the first six digits you entered, enter them again. The monitor takes the last value entered. To exit from the SUBSTITUTE command, press return.

Program counter—When you press "P" on the keyboard, the monitor



Apple PIE + Formatter

Apple PIE (Programma International Editor) and FORMAT (text formatter) offer full strength solutions to today's word processing problems. These versatile, powerful programs provide document preparation and word processing capabilities previously found only on much larger computer systems.

PIE is a general purpose, full screen editor that uses control keys and function buttons to provide a full range of editing capabilities such as search and replace, delete, copy, insert, move. Changes may be made directly anywhere on the screen and are shown as they are performed.

FORMAT uses simple instructions embedded in the input text to describe the desired appearance of the final document. It handles centering, underlining, indenting, page numbering, margins, headers, footers, even form letters, and includes a proofing capability.

These high-quality, cost-effective programs come with comprehensive documentation and run on a 32K Apple II. They are available through your local computer store or direct from Programma International, Inc. at the price of $129.95.

VIDEX VERSION T.M.
DOUBLE VISION T.M.
SUPR TERM VERSION T.M.
STANDARD VERSION

PROGRAMMA
2908 N. Naomi Street
Burbank, Ca. 91504
(213) 954-0240

11

Simple enough for the beginner. Versatile enough for the professional.

Reader Service—see page 178

Photo 9. The completed H-89. Total construction time was about 20 hours.

writes PROGRAM COUNTER on the display, then waits for your next entry. If you hit return, the monitor shows you the current value of the program counter register. At this point, you can enter a new program counter value, or you can hit return to exit.

Go—Pressing "G" will cause the terminal to display GO. You can either enter the address of the first instruction to be executed, or you can just press return, and the H-89 will begin executing at the address already loaded into the program counter register.

Dump—This command lets you save programs on cassette tape. When you press "D," the terminal prints DUMP and waits for you to enter the address range you want stored. One thing that is saved, aside from the program, is the value of the program counter. If, before you enter the DUMP command, you set the program counter value to the first executable instruction location, the program counter will automatically be set to this value when the program is reloaded into the computer.

While the dump is executing, the addresses stored are displayed on the terminal.

Load—The tape load routine reads information back into the H-89 from the cassette tape. If you have several programs on one tape, you must position the tape at the beginning of the file you want to load. The monitor can't search for a specific file. While the file is being loaded, the monitor validates the information coming in. If a checksum error is detected, the tape loading will stop and the terminal will beep until you reset the computer. Once a tape file is properly loaded, just press "G" followed by a return, and the program will begin to execute.

That summarizes the basic command set of the H-89 ROM monitor. A couple of utility routines are available in the monitor, including the memory test. The memory test loads each RAM location with a number, then goes back and reads the number to see if it was stored correctly. The test then increments the original number and begins again. All the while the terminal shows you the number being used in the test.

Another test in the monitor is the disk drive rotational speed test mentioned before. ■

Part two of this article, which focuses on software, will appear in the next issue of Microcomputing.

*Heath now offers a modification to the H-88 or H-89 that will allow CP/M to run, based at address 000H.

This handy program lets your computer convert dollars to English when writing checks.

Dollars and Sense

By Gene Embry

Listing 1. DOLLAR.PRT program.

```
0001 : DOLLAR.PRT
0002 :
0003 : Gene Embry
0004 :
0080 GOSUB 9800::pgm. variables
0099 :
0100 : Main
0101 :
0110 INPUT "Enter a dollar amount  ",N
0112 IF N>9999.99 PRINT"Too large!":GOTO 900
0114 IF N=0 PRINT "Bye!":END
0116 IF N<.01 PRINT "No checks for negative dollars!":GOTO 900
0118 LET N1=N
0120 LET N=INT(N*100)::this makes cents
0122 LET N$=STR$(N)::string it out
0124 PRINT "*** ";
0126 IF N<100 PRINT " zero ";:GOSUB 1000:GOTO 190
0128 LET A$=N$
0132 LET A=LEN(A$)-1
0140 IF A < 1 THEN 190::done
0150 ON A GOSUB 1000,2000,3000,4000,5000
0160 IF F1=1 THEN A=A-1:F1=0
0170 LET A$=RIGHT$(A$,A)
0180 GOTO 132::do it again
0190 PRINT TAB(W);"$ ";N1
0199 :
0900 : Done
0901 :
0910 PRINT
0990 GOTO 100
0999 :
1000 : Cents
1001 :
1010 LET C$=RIGHT$(N$,2)
1020 PRINT "and ";
1030 PRINT C$;"/100 dollars  ***";
1090 RETURN
1099 :
2000 : 0 -> 9 dollars
2001 :
2010 LET B$=LEFT$(A$,1)
2020 LET B=VAL(B$)
2030 IF B=0 THEN 2090
2040 PRINT D$(B);" ";
2090 RETURN
2099 :
3000 : Tens' digit
3001 :
3010 LET B$=LEFT$(A$,1)
3012 LET C$=LEFT$(A$,2)
3020 LET B=VAL(B$)
3022 LET C=VAL(C$)
3030 IF B=0 THEN 3090
3032 IF B+C=0 THEN 3090
```

More

Have you ever wanted to spell out the dollar amount when using your micro to write checks? This routine will let you do just that, using only 27 English words. Add to the vocabulary the words "zero," "dollars," "hundred" and "thousand," and you can print dollar amounts from 1 cent up to $9999.99.

DOLLAR.PRT is shown here as a stand-alone program, but it can be incorporated into your normal check-writing program.

The program is shown in Listing 1, and a sample run, including several examples, is shown in Listing 2.

The Program

Lines 1–100—First, a call is made to routine 9800 to pick up the program's global variables. The array D$( ) is filled with the English words, which are stored as DATA statements in the 9900 section. A word of caution—you must enter the DATA statements in the order shown.

Lines 100–199—This is the main working section. The dollar amount to be converted is entered via line 110. Three limit tests are then performed in lines 112 to 116. You then multiply by 100 so that the amount, N, is now in cents. If the amount is less than $1, you print "zero," call the "cents" routine (1000) and conclude by jumping to line 190. If the amount is greater than $1, you use string (A$) and its length (A) to perform a number of GOSUBs in line

Address correspondence to Gene Embry, Route 1, Box 151-H, Morrisville, NC 27560.

Listing 1 continued.

```
3034 IF C<21 THEN F1=1:GOTO 3050
3040 LET C=20+(B-2)
3050 PRINT D$(C);" ";
3090 RETURN
3099 :
4000 : Hundreds' disits
4001 :
4010 LET B$=LEFT$(A$,1)
4012 LET B=VAL(B$)
4020 IF B=0 THEN 4090
4030 PRINT D$(B);" hundred ";
4090 RETURN
4099 :
5000 : Thousands' disits
5001 :
5010 LET B$=LEFT$(A$,1)
5020 LET B=VAL(B$)
5030 PRINT D$(B);" thousand ";
5090 RETURN
5099 :
9800 : Prosram Variables
9801 :
9810 LINE= 0::Disables the line lensth function
9820 LET D=27::this is # of words needed
9822 DIM D$(D)
9830 FOR X=1 TO D
9832 READ D$(X)::the Enslish words
9834 NEXT X
9840 LET W=60::Position where '$ DDD.CC' is printed
9890 RETURN
9899 :
9900 : Data
9901 :
9910 DATA one,two,three,four,five,six,seven,eisht,nine,ten
9920 DATA eleven,twelve,thirteen,fourteen,fifteen,sixteen
9930 DATA seventeen,eishteen,nineteen,twenty,thirty,fourty
9940 DATA fifty,sixty,seventy,eishty,ninety
```

```
RUN
Enter a dollar amount   ? 1017.56
*** one thousand seventeen and 56/100 dollars  ***          $ 1017.56

Enter a dollar amount   ? .04
***  zero and 4/100 dollars  ***                            $ 0.04

Enter a dollar amount   ? 5.96
*** five and 96/100 dollars  ***                            $ 5.96

Enter a dollar amount   ? 1234.56
*** one thousand two hundred thirty four and 56/100 dollars  ***
                                                            $ 1234.56

Enter a dollar amount   ? -4.77
No checks for nesative dollars!

Enter a dollar amount   ? 999999999
Too larse!

Enter a dollar amount   ? 0
Bye!

BASIC
#
```

Listing 2. Sample run.

150.

Which subroutine is called is based on the value of A. Each call will decrement A by 1. You continue until the value of A is less than 1, and conclude by printing out the cents in line 190.

Example

Assume that the value entered is 1017.56. The value of N will be evaluated as 101756 by line 120. A$ will become 101756, with A taking on the value of 5. Since A is equal to 5, line 150 will cause section 5000, the thousands' digit routine, to be called. The LEFT$ statement in line 5010 assigns a value of 1 to B$. The value of B$ is B, which is 1. Line 5030 looks into the

table D$(B) and finds the English word one. We print D$(B) and append thousand.

Upon return we reassign a new value to A$ via line 170. Its new value is 01756. Then, looping back to line 132, we determine the new value of A is 4 by line 132. Line 150 sends us to routine 4000 to get the English word in the hundreds' position. Since B$=0 and B will equal 0, we return without doing anything.

Line 170 sets A$=1756, and we jump back to line 132. Here the value A becomes 3, thus causing us to go to section 3000 to pick up the tens' digit(s).

Since the digits 11 to 19 are uniquely named in English, we must evaluate both the tens' and units' digits at this point. We determine the sum of both digits (B and C) and then take the necessary measures for printing the word from array D$( ). If F1 gets set to 1 upon return, A is decremented one additional time in line 160.

Finally, A has been set to 0, at which time line 140 will cause the program to continue to line 190, which will print the cents. ■

We're the MAGNOLIA people you've been looking for...

Add the CP/M® disk operating system to your Zenith/Heath '89 All-in-One Computer. Easily installed hardware and software proven by reliable service for more than a year. Supports 8-inch, double-sided 5-inch, and hard disk drives.

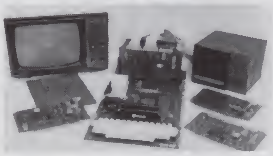
Only $195.

Ask your local dealer, or ✓234

MAGNOLIA MICROSYSTEMS
2812 THORNDYKE AVE. WEST
SEATTLE, WASHINGTON 98199
(206) 285-7266

CP/M® is a registered trademark of Digital Research, Inc.

AIM 65 DEVELOPMENT SYSTEM
DSAIM 65



The DSAIM 65 is a really complete low cost development system for the 6500 microprocessor family. This powerful development system provides the user with a display of 80 characters by 25 lines, 32K bytes of RAM memory expandable to 64K, over 280K bytes of mass storage on line expandable to over 1 Megabyte, 6 card slots expansion board, Centronics-type printer interface, a powerful disk operating system, BASIC high level language, debug monitor, editor, assembler, powerful system library programs for handling of files by same generic names or extension(s).

Items also sold separately
Call or write for further details. ✓334

APPLIED BUSINESS COMPUTER CO.
707 S. State College, Suite G.
Fullerton, Ca. 92631 Tel. 714-871-1411

LEARN

Having trouble learning to use your computer?

Reference manuals don't teach. Most BASIC texts don't cover specific personal computers.

TIS solves these problems with step-by-step books tailored for your machine.

For PET/CBM

Understanding Your PET/CBM $16.95
Vol 1 Basic Programming
PET Graphics .... $ 6.95

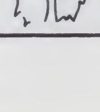
For OSI C1P/C4P

Understanding Your C1P/C4P .... $ 9.95
A Workbook of BASIC Exercises

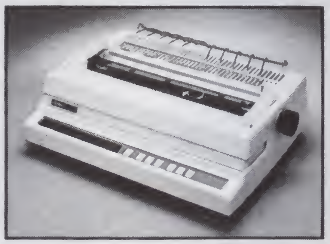
Money Back Guarantee. VISA/MC accepted.

All prices include UPS or 1st Class postage.

TIS, Dept KB ✓95
Box 921
Los Alamos, NM 87544



Data Terminals *Fast* ...from MICROMAIL



DIABLO 630

The Diablo Model 630 is a reliable, high quality, full-character serial printer for anyone who is seeking superior print quality at a low cost. This is the first Diablo printer to offer complete interchangeability between metal and plastic print wheels. And the sophisticated and discerning user does not sacrifice print quality to obtain this versatility. Every aspect of the Diablo 630 design has been focused on maintaining outstanding print quality. Terminals also have self-test, extensive internal diagnostics and automatic bidirectional printing.

$1,999.00

With Adjustable Forms Tractor add **$200.00**

ANADEX DP-9500/9501

The Anadex Models DP-9500 and DP-9501 Alphanumeric Line Printers are designed for all printer applications, including those requiring high density graphics. Standard features include three standard interfaces (RS 232C, Centronics Parallel, and Current Loop), software selectable print sizes including compressed and expanded print, heavy-duty nine-wire printhead (permits true underlining and descending lower case letters), and fast bi-directional printing. The model 9501 offers slightly higher graphics resolution and a slightly slower print speed than the model 9500.

$1,299.00

PRINTERS

ANADEX

DP 9000 **$1199**
DP-9001 **$1199**
Just like the 9500/9501, but 5 inches narrower. Uses paper up to 9.5 inches wide.

T.I.

810/2 **$1599**
(includes upper/lower case option)

810/2 VFC/CP **$1724**
(includes u/l case, forms control & compressed print)

DIABLO

1640 RO **$2525**
Uses plastic daisywheels, prints up to 45 c.p.s.

1650 RO **$2675**
Uses metal daisywheels, prints up to 40 c.p.s.

C.R.T.'s

Tele Video

912C **$ 725**
920C **$ 795**
950 **$ 995**
NEW! Features a detachable keyboard & programmable function keys.

SOROC

IQ120 **$ 689**
IQ140 **$1099**
IQ135 **$ 849**
NEW! Microprocessor controlled, programmable function keys.

C. ITOH

CIT 100 **$1625**

DEC

VT-100 **$1650**

TELEPRINTERS

DEC

LA 34 **$ 969**
Dot-matrix, 30 c.p.s. Adjustable character sizes & line spacing.

LA 34 AA **$1099**
Includes programmable forms length control.

TELETYPE

43 **$ 999**
Very reliable 30 c.p.s. teleprinter. Ideal for use with 300-baud acoustic couplers or modems.

DIABLO

1640 KSR **$2830**
Uses plastic daisywheels, prints up to 45 c.p.s.

1650 KSR **$2940**
Uses metal daisywheels, prints up to 40 c.p.s.

MICROMAIL™ ✓68

MICROMAIL • BOX 3297 • SANTA ANA, CA 92703
(714) 731-4338

To Order: Send check to MICROMAIL, P.O. Box 3297, Santa Ana, CA 92703. Personal or company checks require two weeks to clear. Visa/MasterCard accepted. C.O.D. requires a 15% deposit. **Handling:** Add 3% to orders less than $750, 2% to orders $751 - $2,000, 1% to orders over $2,000. NOTE: Handling charges are waived on orders pre-paid in advance by check. **Shipping:** We ship FREIGHT COLLECT via UPS or Motor Freight. Air and Express delivery is available.

NEW! TPM* for TRS-80 Model II
NEW! System/6 Package
Computer Design Labs

Z80* Disk Software

We have acquired the rights to all TDL software (& hardware). TDL software has long had the reputation of being the best in the industry. Computer Design Labs will continue to maintain, evolve and add to this superior line of quality software.
— Carl Galletti and Roger Amidon, owners.

Software with Manual/Manual Alone

All of the software below is available on any of the following media for operation with a Z80 CPU using the CP/M* or similar type disk operating system (such as our own TPM*).

for TRS-80* CP/M (Model I or II)
for 8" CP/M (soft sectored single density)
for 5¼" CP/M (soft sectored single density)
for 5¼" North Star CP/M (single density)
for 5¼" North Star CP/M (double density)

BASIC I

A powerful and fast Z80 Basic interpreter with EDIT, RENUMBER, TRACE, PRINT USING, assembly language subroutine CALL, LOADGO for "chaining", COPY to move text, EXCHANGE, KILL, LINE INPUT, error intercept, sequential file handling in both ASCII and binary formats, and much, much more. It runs in a little over 12 K. An excellent choice for games since the precision was limited to 7 digits in order to make it one of the fastest around. $49.95/$15.

BASIC II

Basic I but with 12 digit precision to make its power available to the business world with only a slight sacrifice in speed. Still runs faster than most other Basics (even those with much less precision). $99.95/$15.

BUSINESS BASIC

The most powerful Basic for business applications. It adds to Basic II with random or sequential disk files in either fixed or variable record lengths, simultaneous access to multiple disk files, PRIVACY command to prohibit user access to source code, global editing, added math functions, and disk file maintenance capability without leaving Basic (list, rename, or delete). $179.95/$25.

ZEDIT

A character oriented text editor with 26 commands and "macro" capability for stringing multiple commands together. Included are a complete array of character move, add, delete, and display function. $49.95./$15.

ZTEL

Z80 Text Editing Language - Not just a text editor. Actually a language which allows you to edit text and also write, save, and recall programs which manipulate text. Commands include conditional branching, subroutine calls, iteration, block move, expression evaluation, and much more. Contains 36 value registers and 10 text registers. Be creative! Manipulate text with commands you write using Ztel. $79.95/$25.

TOP

A *Z80 Text Output* Processor which will do text formatting for manuals, documents, and other word processing jobs. Works with any text editor. Does justification, page numbering and headings, spacing, centering, and much more! $79.95/$25.

MACRO I

A macro assembler which will generate relocateable or absolute code for the 8080 or Z80 using standard Intel mnemonics plus TDL/Z80 extensions. Functions include 14 conditionals, 16 listing controls, 54 pseudo-ops, 11 arithmetic/logical operations, local and global symbols, chaining files, linking capability with optional linker, and recursive/reiterative macros. This assembler is so powerful you'll think it is doing all the work for you. It actually makes assembly language programming much less of an effort and more creative. $79.95/$20.

MACRO II

Expands upon Macro I's linking capability (which is useful but somewhat limited) thereby being able to take full advantage of the optional Linker. Also a time and date function has been added and the listing capability improved. $99.95/$25.

LINKER

How many times have you written the same subroutine in each new program? Top notch professional programmers compile a library of these subroutines and use a Linker to tie them together at assembly time. Development time is thus drastically reduced and becomes comparable to writing in a high level language but with all the speed of assembly language. So, get the new CDL Linker and start writing programs in a fraction of the time it took before. Linker is compatible with Macro I & II as well as TDL/Xitan assemblers version 2.0 or later. $79.95/$20.

DEBUG I

Many programmers give up on writing in assembly language even though they know their programs would be faster and more powerful. To them assembly language seems difficult to understand and follow, as well as being a nightmare to debug. Well, not with proper tools like Debug I. With Debug I you can easily follow the flow of any Z80 or 8080 program. Trace the program one step at a time or 10 steps or whatever you like. At each step you will be able to see the instruction executed and what it did. If desired, modifications can then be made before continuing. It's all under your control. You can even skip displaying a subroutine call and up to seven breakpoints can be set during execution. Use of Debug I can pay for itself many times over by saving you valuable debugging time. $79.95/$20.

DEBUG II

This is an expanded debugger which has all of the features of Debug I plus many more. You can "trap" (i.e. trace a program until a set of register, flag, and/or memory conditions occur). Also, instructions may be entered and executed immediately. This makes it easy to learn new instructions by examining registers/memory before and after. And a RADIX function allows changing between ASCII, binary, decimal, hex, octal, signed decimal, or split octal. All these features and more add up to give you a very powerful development tool. Both Debug I and II must run on a Z80 but will debug both Z80 and 8080 code. $99.95/$20.

ZAPPLE

A Z80 executive and debug monitor. Capable of search, ASCII put and display, read and write to I/O ports, hex math, breakpoint, execute, move, fill, display, read and write in Intel or binary format tape, and more! on disk

APPLE

8080 version of Zapple

NEW! TPM now available for TRS-80 Model II!

TPM*

A NEW Z80 disk operation system! This is not CP/M*. It's better! You can still run any program which runs with CP/M* but unlike CP/M* this operating system was written specifically for the Z80* and takes full advantage of its extra powerful instruction set. In other words its not warmed over 8080 code! Available for TRS-80* (Model I or II). Tarbell, Xitan DDDC, SD Sales "VERSA-FLOPPY", North Star (SD&DD), and Digital (Micro) Systems. $79.95/$25.

SYSTEM MONITOR BOARD (SMB II)

A complete I/O board for S-100 systems. 2 serial ports, 2 parallel ports, 1200/2400 baud cassette tape interface, sockets for 2K of RAM, 3-2708/2716 EPROM's or ROM, jump on reset circuitry. Bare board $49.95/$20.

ROM FOR SMB II

2KX8 masked ROM of Zapple monitor. Includes source listing $34.95/$15.

PAYROLL (source code only)

The Osborne package. Requires C Basic 2.
5" disks $124.95 (manual not included)
8" disks $ 99.95 (manual not included)
Manual $20.00

ACCOUNTS PAYABLE/RECEIVABLE (source code only)

By Osborne, Requires C Basic 2
5" disks $124.95 (manual not included)
8" $99.95 (manual not included)
Manual $20.00

GENERAL LEDGER (source code only)

By Osborne. Requires C Basic 2
5" disks $99.95 (manual not included)
8" disks $99.95 (manual not included)
Manual $20.00

C BASIC 2

Required for Osborne software. $99.95/$20.

SYSTEM/6

TPM with utilities, Basic I interpreter, Basic E compiler, Macro I assembler, Debug I debugger, and ZEDIT text editor.
Above purchased separately costs $339.75
Special introductory offer: Only $179.75 with coupon!!

$160.

This Coupon Is Worth One Hundred And Sixty Dollars Toward The Full Price Of The SYSTEM/6 Package
System/6 with this coupon is only $179.95.
This is a limited time offer.

$160.00

ORDERING INFORMATION

Visa, Master Charge and C.O.D. O.K. To order call or write with the following information.

1. Name of Product (e.g. Macro I)
2. Media (e.g. 8" CP/M)
3. Price and method of payment (e.g. C.O.D.) include credit card info. if applicable.
4. Name, Address and Phone number.
5. For TPM orders only: Indicate if for TRS 80, Tarbell, Xitan DDDC, SD Sales (5¼" or 8"). ICOM (5¼" or 8"), North Star (single or double density) or Digital (Micro) Systems.
6. N.J. residents add 5% sales tax.

Manual cost applicable against price of subsequent software purchase in any item except for the Osborne software.

For information and tech queries call
609-599-2146

For phone orders ONLY call toll free
1-800-327-9191
Ext. 676
(Except Florida)

OEMS

Many CDL products are available for licensing to OEMs. Write to Carl Galletti with your requirements.

* Z80 is a trademark of Zilog
* TRS-80 is a trademark for Radio Shack
* TPM is a trademark of Computer Design Labs. It is not CP/M*
* CP/M is a trademark of Digital Research

Prices and specifications subject to change without notice.

DEALER INQUIRIES INVITED.

COMPUTER DESIGN LABS
342 Columbus Avenue
Trenton, N.J. 08629

✓18

Add this information to the interpreter described last month, and discover the inner workings of a compiler.

Write Your Own Pseudo-FORTH Compiler

By Richard Fritzson

This compiler, for a simple threaded language similar to FORTH, extends the interpreter for the language described in "Write Your Own FORTH Interpreter" (*Microcomputing*, February 1981, p. 76). The compiler takes reverse Polish notation statements exactly like those typed into the interpreter along with standard programming language control statements such as IF-THEN-ELSE and FOR-LOOP, compiles them into threaded code and stores them in the dictionary associated with a user-defined name.

The compiler is written almost entirely in threaded code and is simply added to the basic interpreter described in the first article, with only one or two simple modifications to the already existing code.

I should point out again that although the language being described is similar to FORTH in many ways, it is not really FORTH. It does not conform to the FORTH implementation standards (see references), nor is it the same language sold by FORTH, Inc. I believe that threaded-code-based systems are so inherently flexible that computerists should use them to devise languages which suit them and to perform software experiments. It would be easy for anyone who is interested to modify the system presented here to conform to a given set of standards.

Language Overview

In this language, every program and subroutine has a name. The traditional name in FORTH for the compiler is a single colon, which introduces a new definition. It is followed by the name of the function being defined, then by the code which comprises the new routine, then finally by a semicolon which terminates the definition. Table 1 contains a few simple examples. (The lowercase text is annotation added after the interaction with the compiler. See the extensions section for suggestions on how to add a comment facility.)

– : ADD + ;	define ADD to be the same as +
0	
– 3 5 ADD	notice the system prints the top of the stack after each interaction
8	
– : R CLEAR 0 ;	define R to CLEAR the stack and push a zero; notice the stack doesn't change until R is executed
8	
– R	
0	
– : ECHOWORD READLINE SCAN POP PRINTS ;	define ECHOWORD to read a line, get one word and print it; notice that definitions may use several lines
0	
– ECHOWORD	executing ECHOWORD; the zero at the end of the echoed word is the value of the stack being printed without a preceding carriage return
HELLO	
HELLO0	
–	typing just a carriage return just gets you the stack value
0	
–	

Table 1. Simple compiler definitions.

While the typed-in language does not let you use the control statements that most programming languages have, such as GOTO, IF-THEN, etc., not much computing can be done without them. This compiler will handle the standard IF-THEN-ELSE statements, as well as two typical looping statements: the FOR LOOP and the REPEAT LOOP (both similar to the Pascal statements of the same name).

Because all of the functions in this language require their arguments to be on the stack, even these standard control statements are written in reverse Polish notation. This will make them seem strange at first. You can either learn to like this or you can write a better compiler. You'll see that doing the latter is not as hard as you might think.

Even though the language here only supports 16-bit integers for data, we will interpret these numbers as logical values when appropriate by using zero to mean false and nonzero to mean true.

The syntax of the IF statement is shown in Table 2. It is the word IF,

Send correspondence to Richard Fritzson, 25 Callodine Ave., Amherst, NY 14226.

condition	IF	word 1	the IF statement without the ELSE
		word 2	
		word 3	
	THEN		
condition	IF	word 1	the IF statement with the ELSE
		word 2	
		word 3	
	ELSE		
		word 4	
		word 5	
	THEN		
– : TEST 5	EQ	IF 1	definition of a trivial function to replace a 5 with a 1
		ELSE 0	and all other numbers with a 0.
		THEN ;	
			EQ is a function which tests the top two elements of the stack and leaves a "true" if they were equal and "false" if not

Table 2. The IF-ELSE-THEN statement.

preceded by code which calculates the condition to be tested, and followed by code to be executed when the condition is true. The code is terminated by "THEN." Optionally, the word ELSE can be inserted before the THEN, followed by code to be executed only when the condition is false.

The "condition" is the top of the stack when the IF is executed. IF pops and examines the top of the stack. If it is false (zero), control transfers to the next instruction after the matching ELSE if there is one, or after the matching THEN if there is no ELSE. The ELSE statement, when encountered during the execution of *the code following* the IF, causes a jump to the matching THEN statement. Examples are shown in Table 2. Again, the syntax is a bit unusual, but it does follow in a reasonable way from the need to use reverse Polish notation.

The syntax for the REPEAT statement is shown in Table 3. The word REPEAT is followed by the code to be repeated, terminated by the word LOOP. At any point between the REPEAT and the LOOP you can put an UNTIL test which, like IF, pops and examines the top of the stack. If it is true (nonzero), control transfers to the instruction following the LOOP word, exiting the loop. If it is false (zero), execution continues with the next instruction following the UNTIL.

Putting the UNTIL immediately prior to the LOOP simulates the action of Pascal's REPEAT-UNTIL loop. Putting it at the beginning of the loop and preceding it with a logical NOT statement simulates Pascal's WHILE-DO loop. Putting the UNTIL anywhere else gives you a flexible, yet still structured, loop instruction. Examples are shown in Table 3.

The FOR statement (Table 4) gives you the indexed iteration found in most programming languages (e.g., the FOR-NEXT loop in BASIC and the DO LOOP in FORTRAN). When FOR is executed, it pops the top value of the stack and uses it as the starting value of the index variable; the second element of the stack is used as the final value. All of the code between the FOR and LOOP words is executed once for each value of the index variable. (Note that in this version the starting and ending values are both positive, and the only increment possible is one.)

Nested FOR LOOPs are allowed to a depth determined by the size of an independent *loop stack* reserved for holding the looping variables. The current value of the index variable is

REPEAT	typical REPEAT LOOP
word1	Words 1, 2, 3, 4 and 5 are executed repeatedly until
word2	the top of the stack after word 3 is nonzero.
word3 UNTIL	
word4	
word5	
LOOP	
– : GETWORD	define GETWORD to READLINEs until SCAN finally
REPEAT	returns a word. Remember that SCAN returns true at
SCAN UNTIL	the top of the stack when it succeeds in finding a
READLINE	word.
LOOP ;	
0	
– : READ# REPEAT	READ# keeps on reading words until one of them can
GETWORD	be successfully converted to a number
CONAXB UNTIL	
POP	
LOOP ;	
0	
– READ#	test READ#
hello	(it just ignores non-numbers)
ss34	
123	
123	(and leaves the number at top)
– : SUM REPEAT	SUM adds all numbers on the stack until it encounters a zero
SWAP	-get second number and duplicate it
DUP	-quit looping if it is zero
EQZ UNTIL	-quit looping if it is zero
+	-else add it to next element
LOOP	-and loop
POP ;	-drop the zero leaving the SUM
0	
– 0 1 2 3 4 5 SUM	test SUM
15	
– 0 1 3 5 7 9 SUM	test SUM
25	

Table 3. The REPEAT LOOP.

```
last first FOR                      The FOR LOOP executes repeatedly for index
        word1                       values from first to last
        word2
        word3
      LOOP

FOR I = first TO last               It is equivalent to the BASIC FOR NEXT loop.
        .
        .
        .
NEXT I

- : SIGMA                           SIGMA adds the numbers from 1 up to its argu-
        0 SWAP                      ment by starting with zero and adding "I" inside a
        1 FOR                       FOR LOOP
          I +
          LOOP ;

0
- 3 SIGMA                           test SIGMA
6
- 10 SIGMA                          test SIGMA
55
- : 2**N                            2**N calculates 2 to the N, where N is the top of
                                    the stack
        1                      (starting value = 2**0)
        SWAP 1 FOR             (repeat from 1 to N)
                DUP +          (double the number)
               LOOP ;

55
- 5 2**N                            test 2**N
32
- 10 2**N                           test 2**N
1024
-
```

Table 4. The FOR LOOP.

accessible through the functions I, J and K. "I" puts the value of the index variable on the stack. "J" is used in a nested pair of loops and puts the value of the outer LOOP's index variable on the stack. "K" is used for triply nested FOR LOOPs and puts the value of the outermost (the third) LOOP's index variable onto the stack. Table 4 contains some examples.

The entire language is built of subroutine calls or function names, so of course there is no explicit subroutine CALL or GOSUB statement. There is no GOTO out of respect for the principles of structured programming. These three control structure statements should let you do anything you need. If you really want a GOTO statement, adding one is a little tricky, but possible and maybe worth trying.

One other feature of the language is that it permits recursive definitions of subroutines. Since parameters, local variables, looping indices and return addresses are all on stacks, these recursive definitions will all work just fine. Table 5 contains some examples; I sometimes find them easier to write than nonrecursive statements. However, you must remember that recursive programs consume return stack space, so you cannot use them to replace looping all of the time (unless you have a lot of return stack space).

```
                                          the recursive definition of SUM does the same
                                          thing as the iterative: it adds the numbers on the
- : SUM                                   stack until it encounters a zero

    SWAP DUP EQZ IF                   (test if second element is a zero)
                   POP                (if yes, then drop it)
                 ELSE
                   +                  (else add the top two numbers)
                   SUM                (and SUM the result and the rest)
                 THEN ;

0
- 0 4 3 5 5 7 SUM                         test recursive SUM
24
- : GETWORD                               a recursive definition of GETWORD
    SCAN EQZ IF                       (scan current line, if scan fails)
         READLINE                     (then READLINE)
         GETWORD                      (and call GETWORD)
       THEN ;

24
-
```

Table 5. Some simple recursive definitions.

Compiler Design

To make the compiler easy to modify and expand, it is as structured as possible. The operation of the main compiler routine is described by the algorithm in Table 6.

For most words of the language—the names of the subroutines in the dictionary—the only compiler action is to store the address of the subroutine in the compiled code (since that is all there is to threaded code). When number constants appear inside a definition, the compiler stores code that, when executed, pushes the number onto the stack (the same way the interpreter pushes numbers on the stack when you type them in). Of course, numbers that are defined as constants in the dictionary are handled in the same way as other entries in the dictionary; the compiler just stores the address of the entry. All of the special compiler language words such as IF, REPEAT, semicolon, etc., are stored in a special dictionary, and the action taken for each is defined as a separate subroutine associated with the dictionary entry.

The organization of the separate dictionary is shown in Table 7. Although the names of the compiler's words are separate from the names of

Now NRI takes you inside the new TRS-80 Model III microcomputer to train you at home as the new breed of computer specialist!

NRI teams up with Radio Shack advanced technology to teach you how to use, program and service state-of-the-art microcomputers…

It's no longer enough to be just a programmer or a technician. With microcomputers moving into the fabric of our lives (over 200,000 of the TRS-80™ alone have been sold), interdisciplinary skills are demanded. And NRI can prepare you with the first course of its kind, covering the complete world of the microcomputer.

Learn At Home in Your Spare Time

With NRI training, the programmer gains practical knowledge of hardware, enabling him to design simpler, more effective programs. And, with advanced programming skills, the technician can test and debug systems quickly and easily.

Only NRI gives you both kinds of training with the convenience of home study. No classroom pressures, no night school, no gasoline wasted. You learn at your convenience, at your own pace. Yet you're always backed by the NRI staff and your instructor, answering questions, giving you guidance, and available for special help if you need it.

Training includes new TRS-80 Model III microcomputer, solid state volt-ohm meter, digital frequency counter, and the NRI Discovery Lab with hundreds of tests and experiments.

(TRS-80 is a trademark of the Radio Shack division of Tandy Corp.)

You Get Your Own Computer to Learn On and Keep

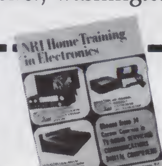
NRI training is hands-on training, with practical experiments and demonstrations as the very foundation of your knowledge. You don't just program your computer, you introduce and correct faults…watch how circuits interact…interface with other systems…gain a real insight into its nature.

You also build test instruments and the NRI Discovery Lab, performing over 60 separate experiments in the process. You learn how your trouble-shooting tools work, and gain greater understanding of the information they give you. Both microcomputer and equipment come as part of your training for you to use and keep.

Send for Free Catalog… No Salesman Will Call

Get all the details on this exciting course in NRI's free, 100-page catalog. It shows all equipment, lesson outlines, and facts on other electronics courses such as Complete Communications with CB, TV and Audio, Digital Electronics, and more. Send today, no salesman will ever bother you. Keep up with the latest technology as you learn on the latest model of the world's most popular computer. If coupon has been used, write to NRI Schools, 3939 Wisconsin Ave., Washington, D.C. 20016.

NRI Schools
McGraw-Hill Continuing Education Center
3939 Wisconsin Avenue
Washington, D.C. 20016
NO SALESMAN WILL CALL
Please check for one free catalog only.



All career courses approved under GI Bill.
☐ *Check for details.*

- ☐ Computer Electronics Including Microcomputers
- ☐ TV/Audio/Video Systems Servicing
- ☐ Complete Communications Electronics with CB • FCC Licenses • Aircraft, Mobile, Marine Electronics
- ☐ CB Specialists Course
- ☐ Digital Electronics • Electronic Technology • Basic Electronics
- ☐ *Small Engine Repair*
- ☐ Electrical Appliance Servicing
- ☐ Automotive Mechanics
- ☐ Auto Air Conditioning
- ☐ Air Conditioning, Refrigeration, & Heating including Solar Technology

Name (Please Print) Age

Street

City/State/Zip

Accredited by the Accrediting Commission of the National Home Study Council 172-031

```
1) Set Compiler Flag to zero
2) Read the Function Name
3) Store the Name and Code pointer in the NAMES section of the dictionary
4) Store a threaded code call (TCALL) as the first word of the compiled definition
5) REPEAT
       Get the Next Word (ignoring line boundaries)
       IF it is a compiler word (e.g., IF, REPEAT, etc.)
       THEN
              Execute it
       ELSE
              IF it is defined in the dictionary
              THEN
                     Store its address in the code
              ELSE
                     IF it is a number
                     THEN
                            store a push instruction
                            store the number
                     ELSE
                            Error—abort compilation
   UNTIL the Compiler Flag is not zero
```

Table 6. The compiler algorithm.

the interpreter's words, the code for both types of entries is intermixed. The compiler part of the dictionary is stored at the top of memory to let the interpreter expand the other part more easily. It would not be difficult to write a program that moved these dictionary parts around and allowed you to expand the compiler (adding, for example, new control statements) in the same way you expand the interpreter.

One reason the compiler is so simple is that it does not do any syntax checking. This means that most syntactic errors will simply cause the compiler to generate bad code. One simple check that is not too difficult to add is to make sure that all of the REPEATs are balanced with UNTILs and LOOPs, that every FOR has a LOOP, that every IF has a THEN and every ELSE has a matching IF-THEN. This only requires that you keep count of the different statements.

A more interesting and more difficult check would be to see to it that no control statement crossed over the boundary of another. For instance, no FOR LOOP can begin inside an IF statement but end outside. Notice that you can't make a REPEAT LOOP begin inside a FOR LOOP but end outside; the compiler will interpret it in the correct way.

As we'll see, compiling code for the IF-THEN-ELSE and REPEAT-LOOP is easy; they're just conditional jumps. The FOR-LOOP, however, requires a bit of work because there is so much to be done for one. The first time the FOR is executed, the numbers on the parameter stack have to be moved to the special loop stack. On each iteration the index variable (on the loop stack) has to be incremented and compared with the final value (also on the loop stack). If it exceeds the final value, the numbers have to be cleared off the loop stack and control transferred to the next instruction after the LOOP.

This whole set of operations is so common to computer programs that some computers have hardware instructions for it (e.g., the Z-80 has a "decrement B and jump if the result is not zero") to make things easy for the compiler and to speed up execution of the loop. For just those reasons we handle this by adding to our threaded code interpreter two new primitives: one for initializing a FOR LOOP and one for incrementing and testing the index variable.

Implementation: The Main Compiler Routine

Listing 1a contains the main compiler routine as it would look if it

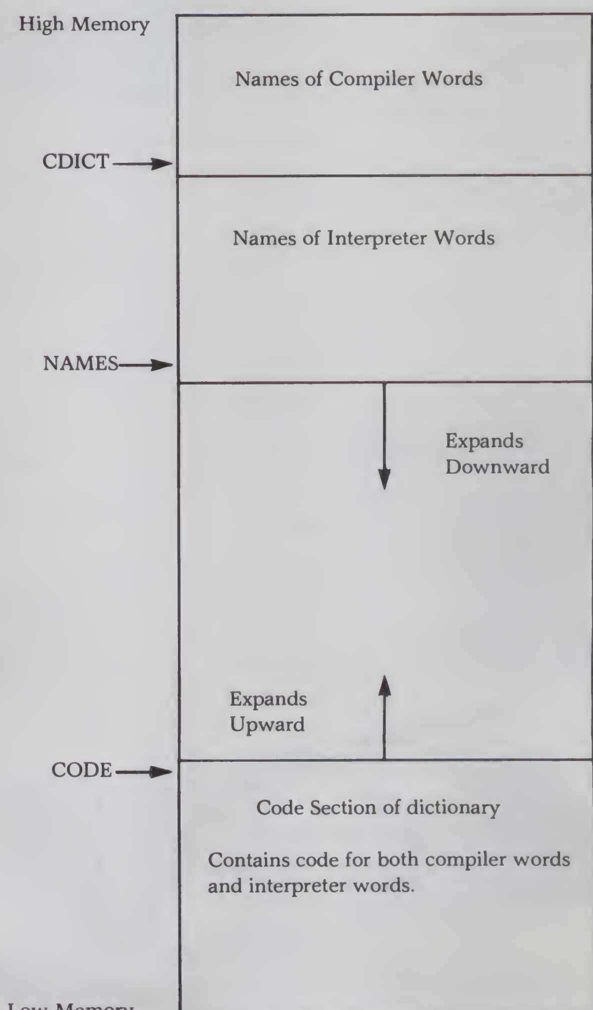


Table 7. Organization of memory. For dictionary details see Fig. 15 of part 1.

```
- : :                                   redefining the colon
  0 CFLAG POKEW                         set compiler flag to zero
  GETWORD                               read the function name
  ENTER                                 and enter it into the dictionary
  @TCALL CMPLW                          compile the address of the TCALL code
                                        (NOTE: @TCALL is a CONSTANT; see
                                        text)

  REPEAT
     GETWORD                            get next function word
     CDICT PEEKW                        get start of compiler word dictionary
     LOOKUP                             and lookup the word
     IF
        EXECUTE                         if found, execute it
     ELSE                               else
        NAMES PEEKW                     look in the interpreter's dictionary
        LOOKUP
        IF
           CMPLW                        if found, compile the word into the code
        ELSE
           CONAXB                       else try interpreting it as a number
           IF
              @PUSH CMPLW               if successful, store a PUSH instruction
                    CMPLW               and the number itself
           ELSE
              CERROR                    otherwise issue an error
           THEN
        THEN
     THEN
     CFLAG PEEKW UNTIL                  check the compiler flag
  LOOP                                  and continue compiling until it is true
```

Listing 1a. The main compiler routine. This function was not actually typed in. Not all of the words used here are defined in the code. It is for illustrative purposes only. However, there is no reason that it cannot be made to work.

Listing 1b. Interactive threaded code compiler.

```
                ; Interactive Threaded Code Compiler
                ;
                ; This is the interactive compiler routine.  Its
                ; name in the dictionary is ":" (a colon).  It
                ; is called with the following syntax:
                ;
                ; - : NAME FN1 FN2
                ; -        FN3 FN4 ... FNN ;
                ;
                ; where name is the name of the new function being
                ; compiled and FN1 through FNN are the
                ; words which make up the new function.  The
                ; final semicolon terminates the definition.
                ;       This code is not optimized, but is exactly
                ; what would be produced if it were compiled by
                ; itself.

                COMPILE
03C3 4702                 DW      TCALL                   ;threaded code CALL
03C5 6C034703BD           DW      ZERO,CFLAG,POKEW        ;clear the compile flag
03CB 53035C05             DW      GETWORD,ENTER           ;get and store name
03CF C7024702B3           DW      TPUSH,TCALL,CMPLW       ;compile threaded call

03D5 5303       COMPL1    DW      GETWORD                 ;get next word
03D7 6804AB022E           DW      CDICT,PEEKW,LOOKUP      ;look in compiler dict
03DD 0503E603             DW      IFZ,COMPL2-1            ;if not found
03E1 9104                 DW      EXECUTE                 ;else execute
03E3 F5021004             DW      JUMP,COMPL5-1           ;and continue

03E7 6004AB022ECOMPL2     DW      NAMES,PEEKW,LOOKUP      ;look in interpreter dict
03ED 0503F603             DW      IFZ,COMPL3-1            ;if not found
03F1 B305                 DW      CMPLW                   ;else store in definition
03F3 F5021004             DW      JUMP,COMPL5-1           ;and continue

03F7 8508       COMPL3    DW      CONAXB                  ;try interpreting as a num
03F9 05030804             DW      IFZ,COMPL4-1            ;if not
03FD C702C702B3           DW      TPUSH,TPUSH,CMPLW       ;else compile code to
0403 B305                 DW      CMPLW                   ;push the number
0405 F5021004             DW      JUMP,COMPL5-1           ;and continue

0409 A307       COMPL4    DW      PRINTS                  ;print the string
040B C7021F04             DW      TPUSH,CUMSG             ;push error message
```

More

Listing 1b continued.

```
040F EB05               DW      UERROR            ;compiler error routine

0411 4703AB02  COMPL5   DW      CFLAG,PEEKW       ;check compiler flag
0415 18031C04           DW      IFNZ,COMPL6-1     ;if not zero, quit
0419 F502D403           DW      JUMP,COMPL1-1     ;else continue

041D 5B02      COMPL6   DW      TRET              ;return when done

041F 0E20697320CUMSG    DB      14,' is undefined.'
                        PAGE
```

```
                ; Top Level External Interpreter Version 1.0
                ;
                ; This routine reads one line of reverse
                ; polish notation from the console and executes it.
                ;
                ; Version 1.5: The code has been modified to provide
                ; LOOKUP with the address of the beginning of the
                ; interpreter dictionary.
                INTERACT
0381 4702               DW      TCALL           ;threaded code

0383 9507               DW      PROMPT          ;prompt the user and
0385 9904               DW      READLINE        ;read a console line

0387 E407      SLOOP    DW      SCAN            ;scan for next word
0389 0503AC03           DW      IFZ,EXIT-1      ;if end of line, quit
038D 6004AB02           DW      NAMES,PEEKW     ;else lookup word in
0391 2E04               DW      LOOKUP          ;interpreter dictionary
0393 05039C03           DW      IFZ,NUMBER-1    ;if not found, try number

0397 9104               DW      EXECUTE         ;else execute it
0399 F5028603           DW      JUMP,SLOOP-1    ;and continue scanning

039D 8508      NUMBER   DW      CONAXB          ;try converting to binary number
039F 18038603           DW      IFNZ,SLOOP-1    ;if succesful, leave on stack
                                                ;and continue scanning

03A3 C702B503           DW      TPUSH,ERRMSG    ;else push error message
03A7 A307               DW      PRINTS          ;and print it
03A9 A307               DW      PRINTS          ;then print string
03AB 5B02               DW      TRET            ;and return

03AD E5023108  EXIT     DW      DUP,CONBXA      ;copy and convert top of stack
03B1 A307               DW      PRINTS          ;print it
03B3 5B02               DW      TRET            ;return

03B5 0D4E6F7420ERRMSG DB        13,'Not Defined: '
                        PAGE
```

Listing 2a. Top level external interpreter, version 1.0.

```
                ; LOOKUP - the dictionary lookup routine
                ; ENTRY: TOP   - pointer to top of dictionary
                ;        TOP-1 - pointer to string to be looked up
                ; EXIT:  TOP   - -1 if string found in dictionary
                ;                 0 if string not found
                ;        TOP-1 - pointer to code of found subroutine
                ;                       or
                ;                string pointer if not found
                ; DESCRIPTION: performs a linear search of the
                ; dictionary. Returns the code address if the string
                ; is found, or else the string pointer if not found.
                ;
                ; Version 1.5: This routine has been modified to
                ; accept the dictionary address as a parameter.
042E 4702      LOOKUP   DW      TCALL           ;threaded code

0430 E502A102  SEARCH   DW      DUP,PEEKB       ;get character count of next entry
0434 05034B04           DW      IFZ,FAIL-1      ;if end of dictionary

0438 6C04               DW      MATCH           ;else attempt a match
043A 18035104           DW      IFNZ,SUCCEED-1  ;if succesful match

043E C1077E02           DW      FIRST,TADD      ;else skip string
0442 C70202007E         DW      TPUSH,2,TADD    ;and pointer
0448 F5022F04           DW      JUMP,SEARCH-1   ;and try next entry

044C D702      FAIL     DW      TPOP            ;drop dictionary pointer
044E 6C03               DW      ZERO            ;leave a zero on the stack
0450 5B02               DW      TRET            ;and quit

0452 DD02D702  SUCCEED  DW      SWAP,TPOP       ;drop string pointer
0456 C1077E02AB         DW      FIRST,TADD,PEEKW ;get code pointer
045C 7403               DW      NEGONE          ;push a minus one
045E 5B02               DW      TRET            ;and return
                        PAGE
```

Listing 2b. The dictionary lookup routine.

were typed into itself. Listing 1b shows what it would be compiled into and how it is actually written in the assembly-language kernel of the interpreter.

In order to let us use the LOOKUP function written for the interpreter, a small modification needs to be made to it. This change makes the beginning of the dictionary a parameter of the LOOKUP routine. The interpreter has to be changed to provide this parameter on the stack. Listing 2 highlights these changes. Since each section of the dictionary ends with a zero byte, the LOOKUP function will not run on beyond the end of the interpreter's section of the dictionary into the compiler's section.

Listing 3 contains the function ENTER, which stores a string into the interpreter's dictionary. It assumes that the code address for the dictionary entry is the current end of the code section of the dictionary because that is where the next compiled definition will go. Notice that the two halves of the dictionary—the names section and the code section—are allowed to grow toward each other until they meet just as described in the interpreter article; testing for this collision is performed both in ENTER and in the routine for adding code to the code section of the dictionary.

Implementation: The Semicolon

The semicolon is in FORTH what the END statement is in BASIC; it tells the compiler that the subroutine definition is finished. The actions taken by ";" (Listing 4) are: 1) a threaded code return (TRET) is stored at the end of the code being compiled, and 2) a flag is set, causing the compiler to exit its main loop.

Implementation: IF-ELSE-THEN

The IF statement is a conditional forward GOTO. When an IF statement is compiled, it stores a conditional jump-if-zero (i.e., *skip this code if false*) instruction (a threaded code jump instruction, not a machine-language instruction) but leaves the address to be jumped to as a zero temporarily. *The address of the word which contains this zero* (the second half of the jump instruction) *is pushed onto the stack* so that it can be used later. Listing 5 contains all of the code for the IF, THEN and ELSE statements. It is written in threaded code.

The THEN statement serves as a target for the forward jump left by the IF. The compiler routine for

✓Reader Service—see page 178

```
                ; ENTER - enter a name in the dictionary
                ; ENTRY:  TOP   - pointer to name
                ; EXIT:   No Values Returned
                ; DESCRIPTION: Moves the name up to the name section of
                ; the interpreter's dictionary.  Produces an error
                ; if there is no room.
055C 4702       ENTER   DW      TCALL           ;threaded code
055E 6404AB02           DW      CODE,PEEKW      ;store code address for
0562 6004AB02           DW      NAMES,PEEKW     ;new dictionary entry
0566 76027602BD         DW      DEC,DEC,POKEW   ;(in word below current name point
056C E502A102           DW      DUP,PEEKB       ;get the string length
0570 E5026E026E         DW      DUP,INC,INC,INC ;add 3 giving entry length
0578 6004AB02           DW      NAMES,PEEKW     ;subtract from current beginning
057C DD029902           DW      SWAP,TSUB       ;of NAMES giving new beginning
0580 E5026404AB         DW      DUP,CODE,PEEKW  ;compare with CODE
0586 DD02330399         DW      SWAP,IFGT,EERROR-1 ;if memory full, then error
058C E5026004BD         DW      DUP,NAMES,POKEW ;else store new beginning
0592 DD026E02F7         DW      SWAP,INC,MVBYTES ;and move string to it
0598 5B02               DW      TRET

059A C702A205   EERROR  DW      TPUSH,EEMSG     ;print bad news
059E D905               DW      ERROR
05A0 5B02               DW      TRET

05A2 0F44696374EEMSG    DB      15,'Dictionary Full',0DH
                        PAGE
```

Listing 3. Enter a name in the dictionary.

```
                ; SEMI - the semicolon - end compilation
                ; DESCRIPTION: compiles a threaded code return and sets
                ; the compiler flag to -1.

06A3 4702       SEMI    DW      TCALL           ;threaded code
06A5 C7025B02B3         DW      TPUSH,TRET,CMPLW ;compile a TRET
06AB 74034703BD         DW      NEGONE,CFLAG,POKEW ;signal end of compilation
06B1 5B02               DW      TRET
                        PAGE
```

Listing 4. Code for the semicolon.

```
                ; Compiler Vocabulary

                ; IF - skip code if false
                ; EXIT:  TOP - address of middle of jump instruction
                ; DESCRIPTION: compiles a jump-if-zero saving
                ; the address of the destination address on the
                ; stack.

060D 4702       IFX     DW      TCALL           ;threaded code
060F C7020503B3         DW      TPUSH,IFZ,CMPLW ;compile conditional jump
0615 6404AB02           DW      CODE,PEEKW      ;save next address
0619 6C03B305           DW      ZERO,CMPLW      ;compile a zero temporarily
061D 5B02               DW      TRET

                ; THEN - target for IF and ELSE jumps
                ; ENTRY: TOP   - Address of second half of jump
                ;                instruction to be modified
                ; DESCRIPTION: stores the address of the next code
                ; word to be compiled in the jump instruction addressed
                ; by the top of the stack.

061F 4702       THEN    DW      TCALL           ;threaded code
0621 6404AB0276         DW      CODE,PEEKW,DEC  ;get next CODE address
0627 DD02BD02           DW      SWAP,POKEW      ;store in jump instruction
062B 5B02               DW      TRET

                ; ELSE - unconditional skip to THEN; target for IF jumps
                ; ENTRY: TOP   - Address of second half of jump
                ;                instruction to be modified.
                ; EXIT:  TOP   - Address of second half of jump
                ;                instruction to be modified (new jump)
                ; DESCRIPTION: compiles and unconditional jump saving
                ; the address of the destination address on the stack.
                ; Stores the address of the next code word to be compiled
                ; in the jump instruction addressed by the top of the stack.

062D 4702       ELSEX   DW      TCALL           ;threaded code
062F C702F502B3         DW      TPUSH,JUMP,CMPLW ;compile unconditional jump
0635 6404AB02           DW      CODE,PEEKW      ;save address of next word
0639 6C03B305           DW      ZERO,CMPLW      ;compile a zero temporarily
063D DD02               DW      SWAP            ;get address of IF jump
063F 6404AB0276         DW      CODE,PEEKW,DEC  ;get next CODE address
0645 DD02BD02           DW      SWAP,POKEW      ;store in IF jump
0649 5B02               DW      TRET
                        PAGE
```

Listing 5. Compiler vocabulary.

THEN does not compile any new code, but instead uses the address left on the stack by IF to complete the forward jump by storing there the address of the next word of code to be compiled.

The ELSE statement uses a combination of the IF code and the THEN code. It completes the IF's forward jump in the same way as the THEN statement—so that when an ELSE is present, IF only jumps as far as the ELSE and not all the way to the THEN. However, in order to make the "true" code skip over the "false" code, ELSE also compiles an unconditional jump and leaves the address of the second word of the jump on the stack. In this way, the IF and the ELSE leave identical stack conditions, and the THEN doesn't have to worry about whether the optional ELSE appeared or not.

Implementation: REPEAT - UNTIL - LOOP

The conditional looping routines work in almost the same way as the IF-ELSE-THEN routines. The REPEAT statement is only a target for a backwards jump from the LOOP statement. It doesn't compile any code, but simply leaves the address of the next word to be compiled on the stack. The UNTIL statement, which is not optional, compiles a conditional jump-if-not-zero and, like the IF statement, leaves the address to jump to as zero and the address of the second half of the jump on the stack.

When the LOOP statement is executed, there are two words on the stack: the address of the second half of the forward jump (left by UNTIL) and the address of the beginning of the loop (left by REPEAT). The LOOP routine compiles an unconditional jump back to the REPEAT statement and completes the forward jump of the UNTIL statement so that it jumps past the LOOP. This code is shown in Listing 6.

The use of the stack to store jump addresses is all done during compilation and has no effect on parameter passing at run time. It allows nesting of the IF-THEN, REPEAT *and* FOR constructions to a depth which is limited only by the size of the parameter stack and will work fine as long as each statement is balanced and properly constructed. As mentioned earlier, syntax errors will not cause compiler errors but will result in the generation of bad code as, for example, the THEN statement completing a

```
                ; REPEAT - target for LOOP instruction
                ; EXIT:  TOP   - address of end of CODE
                ; DESCRIPTION: simply leaves address of next CODE
                ; instruction on the stack.

064B 4702       REPEAT  DW      TCALL             ;threaded code
064D 6404AB0276         DW      CODE,PEEKW,DEC    ;get CODE address
0653 5B02               DW      TRET

                ; UNTIL - terminate LOOP if true
                ; EXIT:  TOP   - address of middle of jump instruction
                ;                to be filled in by LOOP.
                ; DESCRIPTION: Compiles a conditional jump-if-not-zero
                ; leaving the address of the destination address on the stack
                ; to be filled in by the LOOP instruction.

0655 4702       UNTIL   DW      TCALL             ;threaded code
0657 C7021803B3         DW      TPUSH,IFNZ,CMPLW  ;compile conditional jump
065D 6404AB02           DW      CODE,PEEKW        ;save address of next word
0661 6C03B305           DW      ZERO,CMPLW        ;compile a zero temporarily
0665 5B02               DW      TRET

                ; LOOP - target for UNTIL; unconditional return to REPEAT
                ; ENTRY: TOP   - address of UNTIL jump
                ;        TOP-1 - address of REPEAT location
                ; DESCRIPTION: compiles an unconditional jump to the
                ; location of the REPEAT word; points the UNTIL jump
                ; to the next CODE word to be compiled.

0667 4702       LOOP    DW      TCALL             ;threaded code
0669 C702F502B3         DW      TPUSH,JUMP,CMPLW  ;compile unconditional jump
066F DD02B305           DW      SWAP,CMPLW        ;to repeat
0673 6404AB0276         DW      CODE,PEEKW,DEC    ;get next CODE word
0679 DD02BD02           DW      SWAP,POKEW        ;and complete UNTIL jump
067D 5B02               DW      TRET
                        PAGE
```

Listing 6. REPEAT—target for LOOP instruction.

jump set up by an UNTIL, or some similar confusion.

Implementation: FOR-LOOP

As mentioned before, to keep the FOR and LOOP code simple, we introduce two new primitive instructions into the threaded code interpreter: the initialize FOR loop instruction (FORINIT) and the increment and test instruction (FORTEST). Both of these use a new interpreter stack (recall that we already have two: the parameter stack and the return address stack), the LOOP stack (Listing 7). It is implemented in the same way as the return stack (the stack pointer is kept in memory) and is used to store the current and final values of the loop variable.

FORINIT, since it is simple and only executed once for each use of the FOR LOOP, is written in threaded code (Listing 8). It pushes onto the loop stack one less than the starting value of the index variable (so that the first increment and test instruction will bump it to its proper starting value) and one more than the final value of the loop variable (so that the test for termination is a test for equality rather than a greater than test).

FORTEST is written in machine code so that it will execute quickly. It pops the two numbers off of the loop stack, increments the index variable and compares it to the terminating value. If they are equal, it leaves a nonzero quantity on the stack; if they are not equal, it pushes both values

```
                ; LPUSHX - push top of stack on loop stack
                ; ENTRY: DE  - number to go to loop stack
                ; EXIT:  No Values Returned
                ; DESCRIPTION: Pushes DE on the loop stack.

0734 2A5307     LPUSHX  LHLD    LSTACK  ;get stack pointer
0737 73                 MOV     M,E     ;store low byte
0738 23                 INX     H       ;bump stack pointer
0739 72                 MOV     M,D     ;store high byte
073A 23                 INX     H       ;bump stack pointer
073B 225307             SHLD    LSTACK  ;restore pointer
073E C9                 RET

                ; LPUSH - push top of stack on loop stack
                ; DESCRIPTION: This is the threaded code callable
                ; version of LPUSHX.

073F 4107       LPUSH   DW      $+2     ;machine code
0741 D1                 POP     D       ;get value
0742 CD3407             CALL    LPUSHX  ;push it
0745 C33702             JMP     NEXT

                ; LPOPX - pop loop stack
                ; ENTRY: No Values Expected
                ; EXIT:  DE - former top of loop stack
                ; DESCRIPTION: Pops loop stack to DE

0748 2A5307     LPOPX   LHLD    LSTACK  ;get stack pointer
074B 2B                 DCX     H       ;dec to hi byte
074C 56                 MOV     D,M     ;retieve
074D 2B                 DCX     H       ;dec to low byte
074E 5E                 MOV     E,M     ;retrieve
074F 225307             SHLD    LSTACK  ;restore stack pointer
0752 C9                 RET

                ; LSTACK - the loop stack
                ; DESCRIPTION: Implemented in the same way as the return stack.
                ; The stack pointer points to the next available location.

0753 5507       LSTACK  DW      $+2
0755                    DS      64      ;good for 16 levels of nested loops
                        PAGE
```

Listing 7. LPUSHX—push top of stack on loop stack.

```
- : CONSTANT
           GETWORD ENTER        read the next word and enter it into the dic-
                                tionary
           CONS     CMPLW       compile a pointer to CONSTANT code and
                    CMPLW ;     compile the constant's value
0
- : VARIABLE
           GETWORD ENTER        read the next word and enter it
           VARBL    CMPLW       compile pointer to VARIABLE code
                    CMPLW;      and compile initial value

0
- 10 CONSTANT TEN               define TEN to be the constant 10
0
- TEN
10
- TEN TEN +
20
- 13 VARIABLE UNLUCKY           define UNLUCKY to be a variable with the ini-
20                              tial value 13
- UNLUCKY                       variables return their addresses
2433
- UNLUCKY PEEKW                 you have to "peek" their values
13
- 7 UNLUCKY POKEW               and "poke" changes to them
13
- UNLUCKY PEEKW
7
- : ? PEEKW ;                   of course you don't have to be verbose about it
7
- UNLUCKY ?
7
-
```

Table 8. CONSTANTS and VARIABLES.

back on the loop stack and leaves a zero on the stack.

These two primitives make the compilation of the FOR LOOP quite simple. FOR compiles a call to FORINIT and a call to FORTEST, leaving the address of the FORTEST instruction on the stack to serve as a target for the jump from LOOP. It also compiles a jump-if-not-zero after the FORTEST and leaves the address of the second half of the jump on the stack (Listing 9).

Notice that the stack conditions left by FOR are the same as those left by REPEAT and UNTIL. Therefore, the same LOOP code can be used for both loops. LOOP compiles an unconditional jump back to the FORTEST instruction and completes the forward jump which follows it with the address of the word to be compiled after LOOP.

Implementation: I, J and K

The instructions which access the current value of the index variable of the FOR LOOP are not compiler instructions. They are regular threaded-code instructions which access the contents of the loop stack; the compiler simply compiles their address. Listing 10 shows their definitions. Notice that they access every other word on the loop stack, skipping the terminating values. It should be clear how to add more of these words if they are needed.

Implementation: CONSTANTS and VARIABLES

The interpreter article introduced a trick for implementing variables and constants in a threaded language; you create an entry with a special type of function which either loads a constant onto the stack or loads the address of a variable. Since, while typing new programs into the compiler, you will probably want to introduce new variables and constants, you need a way to enter these definitions for them.

One method is to provide the

Kilobaud Microcomputing will feature a special word processing issue in May. Manufacturers of word processing software for any microcomputer who want to be included in this issue should contact the *Microcomputing* editoral department immediately. Please send spec sheets of your software system. We will contact you if we need additional information.

FMG CORPORATION NOW CARRIES GRAHAM-DORIAN & PEACHTREE SOFTWARE

FMG CORPORATION

VISA · master charge

P.O. Box 16020
Fort Worth, Texas
76133
(817) 294-2510

70

M-530

All FMG Software Products Include All Necessary Manuals

SOFTWARE / WITH MANUAL · MANUAL/ALONE

· MICROPRO INTERNATIONAL

(M) **SUPER-SORT I** — Sort, merge, extract utility as absolute executable program or linkable module in Microsoft format. Sorts fixed or variable records with data in binary, BCD, Packed Decimal, EBCDIC, ASCII, floating & fixed point, exponential, field justified, etc. Even variable number of fields per record! **$225/$25**

SUPER-SORT II — Above available as absolute program only ..... **$175/$25**

SUPER-SORT III — As II without SELECT/EXCLUDE ..... **$125/$25**

WORD-STAR — Menu driven visual word processing system for use with standard terminals. Text formatting performed on screen. Facilities for text paginate, page number, justify, center and underscore. User can print one document while simultaneously editing a second. Edit facilities include global search and replace, Read/Write to other text files, block move, etc. Requires CRT terminal with addressable cursor positioning ..... **$495/$40**

WORD-STAR Customization Notes — For sophisticated users who do not have one of the many standard terminal or printer configurations in the distribution version of WORD-STAR ..... **NA/$95**

WORD-MASTER Text Editor — In one mode has superset of CP/M's ED commands including global searching and replacing, forwards and backwards in file. In video mode, provides full screen editor for users with serial addressable-cursor terminal ..... **$150/$25**

FLOPPY SAVER — Protection for center holes of 5" and 8" floppy disks. Only 1 needed per diskette. Kit contains centering post, pressure tool and tough 7 mil mylar reinforcing rings for 25 diskettes.

5", Kit	$14.95
5", Rings only	$7.95
8", Kit	$18.95
8", Rings only	$8.95

HEAD CLEANING DISKETTE — Cleans the drive Read/Write head in 30 seconds. Diskette absorbs loose oxide particles, fingerprints, and other foreign particles that might hinder the performance of the drive head. Lasts at least 3 months with daily use.

8"	$32.00
5¼"	$30.00

(M) · **DESPOOL** — Allows flexibility and efficiency. (Disk file printing can be accomplished while simultaneously using the computer for other tasks) Slower printers do not tie up the computer. Requires 32K minimum ...... **$75/$10**

(M) · **SCREEN EDIT** — Text editor for program entry — allows user the ability to see entries as they are being made. Has command which enables user to move the viewed position of the file anywhere within the current data file OR add information anywhere in the file. Requires 16K minimum ..... **$125/$25**
(Also available in TRS DOS format.
Specify model or TRS-80)

(M) · **MAC** — Disk-based, powerful macro assembler utilizes Standard Intel Mnemonics. Includes macro processor.
The CP/M 8080 Macro Assembler reads assembly language statement from a diskette file and produces an Intel "HEX" format object file on the disk suitable for processing in the TRS-CP/M environment. Requires 32K minimum and CP/M ..... **$100/$25**

(M) · **ZSID** — Efficient and reliable program testing system for Z80 microcomputers. Capabilities include traceback and histogram facilities. Allows real time break points.
ZSID is a symbolic debugger which expands upon the features of the TRS-CP/M standard debugger, providing greatly enhanced facilities for assembly language program check-out. Requires 32K minimum and CP/M ...... **$99/$25**

· PEACHTREE SOFTWARE SYSTEMS

(T) **GENERAL LEDGER** — Records details of all financial transactions. Generates a balance sheet and an income statement. Flexible and adaptable design for both small businesses and firms performing client writeup services. Produces reports as follows: Trial Balance, Transaction Registers, Balance Sheet, Prior Year Comparative Balance Sheet, Income Statement, Prior Year Comparative Income Statement and Departmental Income Statements. Interactive with other PEACHTREE accounting packages. Supplied in source code for Microsoft BASIC ..... **$990/$60**

(T) **ACCOUNTS PAYABLE** — Tracks current and aged payables and incorporates a check writing feature. Maintains a complete vendor file with information on purchase orders and discount terms as well as active account status. Produces reports as follows: Open Voucher Report, Accounts Payable Aging Report and Cash Requirements. Provides input to PEACHTREE General Ledger. Supplied in source code for Microsoft BASIC ..... **$990/$60**

(T) **ACCOUNTS RECEIVABLE** — Generates invoice register and complete monthly statements. Tracks current and aged receivables. Maintains customer file including credit information and account status. The current status of any customer account is instantly available. Produces reports as follows: Aged Accounts Receivable, Invoice Register, Payment and Adjustment Register and Customer Account Status Report. Provides input to PEACHTREE General Ledger. Supplied in source code for Microsoft BASIC ..... **$990/$60**

(T) **PAYROLL** — Prepares payroll for hourly, salaried and commissioned employees. Generates monthly, quarterly and annual returns. Prepares employee W-2's. Includes tables for federal withholding and FICA as well as withholding for all 50 states plus up to 20 cities from pre-computed or user generated tables. Will print checks, Payroll Register, Monthly Summary and Unemployment Tax Report. Provides input to PEACHTREE General Ledger. Supplied in source code for Microsoft BASIC ..... **$990/$60**

(T) **INVENTORY** — Maintains detailed information on each inventory item including part number, description, unit of measure, vendor and reorder data, item activity and complete information on current item costs, pricing and sales. Produces reports as follows: Physical Inventory Worksheet, Inventory Price List, Departmental Summary Report, Inventory Status Report, The Reorder Report and the Period-to-Date and Year-to-Date reports. Supplied in source code for Microsoft BASIC ..... **$1,190/$60**

(T) **MAILING ADDRESS** — Keeps track of name and address information and allows the selective printing of this information in the form of mailing lists or address labels. Allows the user to tailor the system to his own particular requirements. User-defined format and print-out system uses a special format file which tells programs how to print the mailing list or address labels. Standard format files are included with system. Automatic sorting of data uses Indexed file management routines which allow the name and address information to be sequentially retrieved and printed without file sorting. Supplied in source code for Microsoft BASIC ..... **$790/$60**

· GRAHAM-DORIAN SOFTWARE SYSTEMS

(T) **GENERAL LEDGER** — An on-line system; no batching is required. Entries to other GRAHAM-DORIAN accounting packages are automatically posted. User establishes customized C.O.A. Provides transaction register, record of journal entries, trial balances and monthly closings. Keeps 14 month history and provides comparison of current year with previous year. Requires CBASIC-2. Supplied in source ..... **$995/$35**

(T) **ACCOUNTS PAYABLE** — Maintains vendor list and check register. Performs cash flow analysis. Flexible — writes checks to specific vendor for certain invoices or can make partial payments. Automatically posts to GRAHAM-DORIAN General Ledger or runs as stand alone system. Requires CBASIC-2. Supplied in source ..... **$995/$35**

(T) **ACCOUNTS RECEIVABLE** — Creates trial balance reports, prepares statements, ages accounts and records invoices. Provides complete information describing customer payment activity. Receipts can be posted to different ledger accounts. Entries automatically update GRAHAM-DORIAN General Ledger or runs as stand alone system. Requires CBASIC-2. Supplied in source ..... **$995/$35**

(T) **PAYROLL SYSTEM** — Maintains employee master file. Computes payroll withholding for FICA, Federal and State taxes. Prints payroll register, checks, quarterly reports and W-2 forms. Can generate ad hoc reports and employee form letters with mail labels. Requires CBASIC-2. Supplied in source ..... **$590/$35**

(T) **INVENTORY SYSTEM** — Captures stock levels, costs, sources, sales, ages, turnover, markup, etc. Transaction information may be entered for reporting by salesman, type of sale, date of sale, etc. Reports available both for accounting and decision making. Requires CBASIC-2. Supplied in source ..... **$590/$35**

(T) **JOB COSTING** — Designed for general contractors. To be used interactively with other GRAHAM-DORIAN accounting packages for tracking and analysing expenses. User establishes customized cost categories and job phases. Permits comparison of actual versus estimated costs. Automatically updates GRAHAM-DORIAN General Ledger or runs as stand alone system. Requires CBASIC-2. Supplied in source **$995/$35**

Sample Program Disk For Each Graham-Dorian Business Package. Specify Package **$45**

The sale of each proprietary software package conveys a license for use on one system only.

Prices F.O.B. Fort Worth, Tex. Shipping, handling and C.O.D. charges extra.

· MICROSOFT PRODUCTS

(M) **BASIC-80** — Disk Extended BASIC, ANSI compatible with long variable names, WHILE/WEND, chaining, variable length file records ..... **$350/$25**

(M) **BASIC COMPILER** — Language compatible with BASIC-80 and 3-10 times faster execution. Produces standard Microsoft relocatable binary output. Includes MACRO-80. Also linkable to FORTRAN-80 or COBOL-80 code modules ..... **$395/$25**

(M) **FORTRAN-80** — ANSI 66 (except for COMPLEX) plus many extensions. Includes relocatable object compiler, linking loader, library with manager. Also includes MACRO-80 (see below) ..... **$500/$25**

(M) **COBOL-80** — Level 1 ANSI '74 standard COBOL plus most of Level 2. Full sequential, relative, and indexed file support with variable file names. STRING, UNSTRING, COMPUTE, VARYING/UNTIL, EXTEND, CALL, COPY, SEARCH, 3-dimensional arrays, compound and abbreviated conditions, nested IF. Powerful interactive screen-handling extensions. Includes compatible assembler, linking loader, and relocatable library manager as described under MACRO-80 ..... **$750/$25**

(M) **MACRO-80** — 8080/Z80 Macro Assembler. Intel and Zilog mnemonics supported. Relocatable linkable output. Loader, Library Manager and Cross Reference List utilities included ..... **$150/$25**

(M) **XMACRO-86** — 8086 cross assembler. All Macro and utility features of MACRO-80 package. Mnemonics slightly modified from Intel ASM86 Compatibility data sheet available ..... **$300/$25**

(M) **PASCAL/MT** — Subset of standard PASCAL. Generates ROMable 8080 machine code. Symbolic debugger included. Supports interrupt procedures, CP/M file I/O and assembly language interface. Real variables can be BCD, software floating point, or AMD 9511 hardware floating point. Version 3 includes Enumeration and Record data types. Manual explains BASIC to PASCAL conversion. Source for the runtime package requires Digital Research's MAC. Requires 32K ..... **$250/$30**

(M) **CBASIC-2** Disk Extended BASIC — Non-interactive BASIC with pseudo-code compiler and run-time interpreter. Supports full file control, chaining, integer and extended precision variables, etc. ..... **$110/$15**

(M) **BSTAM** — Utility to link one computer to another also equipped with BSTAM. Allows file transfers at full data speed (no conversion to hex), with CRC block control check for very reliable error detection and automatic retry. We use it! It's great! Full wildcard expansion to send ★ COM, etc. 9600 baud with wire. 300 baud with phone connection. **Both** ends need one. Standard and (M) versions can talk to one another. ..... **$150/$5**

(M) (T) **SELECTOR III-C2** — Data Base Processor to create and maintain multi Key data bases. Prints formatted sorted reports with numerical summaries or mailing labels. Comes with sample applications, including Sales Activity, Inventory, Payables, Receivables, Check Register, and Client/Patient Appointments, etc. Requires CBASIC-2. Supplied in source **$349/$20**

GLECTOR — General Ledger option to SELECTOR III-C2. Interactive system provides for customized COA. Unique chart of transaction types insure proper double entry bookkeeping. Generates balance sheets, P&L statements and journals. Two year record allows for statement of changes in financial position report. Supplied in source. Requires SELECTOR III-C2, CBASIC-2 and 52K system ..... **$250/$25**

(M) **TEXTWRITER III** — Text formatter to justify and paginate letters and other documents. Special features include insertion of text during execution from other disk files or console, permitting recipe documents to be created from linked fragments on other files. Has facilities for sorted index, table of contents and footnote insertions. Ideal for contracts, manuals, etc. Now compatible with Electric Pencil* prepared files. ..... **$125/$20**

ORDERS MUST SPECIFY DISK SYSTEMS AND FORMATS:

FORMATS AVAILABLE:
(A) TRS-80 Model I (M) Keys Only
(B) TRS-80 Model II
(C) TRS-80 Model III (M) Keys Only
(D) HEATHKIT H89 (M) Keys Only
(E) NORTH STAR
(F) SUPER BRAIN QD
(G) STANDARD UNIMPLEMENTED

(M) Modified version available for use with CPM as implemented on Heath and TRS-80 Model I computers.

(T) For all (T) items listed above, the recommended system configuration consists of 48K CPM 2 full dize disk drives. 24 x 80 CRT and 132 column printer.

NEW VERSATILITY
For Your TRS-80

CP/M®2

CONTROL PROGRAM FOR MICROCOMPUTERS ENABLING YOU TO RUN SOFTWARE PUBLISHED FOR CP/M 1.4 ON THE TRS-80

CP/M is considered the industry standard disk operating system because it gives you the hardware-independent interface you need to make your computer work for you. CP/M 2.0 is the latest in the evolution of a proven reliable and efficient software system. FMG CORPORATION NOW OFFERS THE CP/M 2.0 FOR THE TRS-80.
It features an enhanced upward compatible file system and powerful new random access capabilities. The CP/M 2.0 from FMG provides the ability to run software published for the CP/M system, on the TRS-80 Model II. From minidisks, floppy disks, all the way to high-capacity hard disks, the flexibility of CP/M 2.0 makes it a truly universal operating system. The package includes an 8" system disk, editor, assembler and debugger for the TRS-80

Available in Format A, B, C, G only ..... **$200/$25**

MP/M™

MULTI-PROGRAMMING MONITOR

NEW INDUSTRY STANDARD

A *deluxe operating* system that provides big computer facilities at small computer prices. MP/M is a monitor program which operates with your microcomputer to provide multi-terminal access with multi-programming at each terminal. Best of all, it's CP/M compatible which means you can run a wide variety or programming languages, applications packages, and development software.

You can run simultaneous editors, program translators, and background printer spoolers. Or you can use MP/M for data entry or data-base access from remote terminals. Or you can use MP/M real-time features to monitor an assembly line and automatically schedule programs for execution throughout the day. MP/M makes an excellent focal point for a cluster of connected microcomputers. The possibilities are limitless.

(Format B)	$450/$35
(Format G)	$300/$35

*CP/M and MP/M are trademarks of Digital Research.
Z80 is a trademark of Zilog, Inc.
TRS-80 is a trademark of Tandy Corp
Pascal/M is a trademark of Sorcim.

FMG's LIBRARY:

PASCAL USER MANUAL & REPORT
(2nd) Edition by K. Jensen and N. Wirth

- **Tutorial Manual and Concise Reference Report for Both Programmers and Implementors**
- **Includes Helpful Examples to Demonstrate the Various Features of PASCAL**

The book consists of two parts: the user manual and the revised report. The manual is directed to those who have some familiarity with computer programming and who wish to get acquainted with the PASCAL language. The report defines standard PASCAL, which constitutes a common base between various implementations of the language

Stock No. #B21 — **Price** $9.95

PASCAL PRIMER Problem Solving

This book has three major goals:
- **To introduce all aspects of the programming and problem solving process (includes problem specification and organization, algorithms, coding, debugging, testing, documentation and maintenance);**
- **To teach good programming style and how to produce a high quality finished product; and**
- **To teach the syntax of the PASCAL programming language.**

Numerous examples are employed throughout the text. PASCAL is used as a vehicle to teach various aspects of programming techniques

Stock No. #B24 — **Price** $18.95

Microcomputer Problem Solving Using Pascal
by Kenneth L. Bowles

- **A Book Designed for Both College Courses AND Individual Self-Study**
- **Ideal for use with UCSD Pascal**
- **Includes Extensions to Standard PASCAL**

This book is designed both for introductory courses in computer problem solving at the freshman and sophomore college level, and for individual self-study. It includes many examples and actually executable programs. It includes information on the necessary functions and procedures for handling graphics and strings

Stock No. #B22 — **Price** $14.95

BEGINNER'S MANUAL FOR UCSD PASCAL SYSTEM

- **An Enlightening Introduction to UCSD PASCAL**
- **Demonstrates How to Use the UCSD PASCAL System and How to Program in PASCAL**
- **Includes Many Practical Examples of PASCAL Programs**

This book is intended to be used as an introduction and reference manual for people just beginning to use the UCSD Pascal Software System. Whether you have never used a computer before or whether you are an experienced programmer who is unfamiliar with UCSD PASCAL, this book will provide a relatively easy, yet thorough, introduction to UCSD PASCAL

Stock No. #B25 — **Price** $11.95

PROGRAMING IN PASCAL
by Peter Grogono

- **An Excellent Introduction to One of the Fastest Growing Programming Languages Today**
- **Sections on Procedures and Files PLUS a Chapter on Dynamic Data Structures such as Trees and Linked Lists**

The text is arranged as a tutorial, containing both examples and exercises to increase reader proficiency in PASCAL. Concepts are illustrated by examples, ranging from the Tower of Hanoi problem to circumscribing a circle about a triangle. PROGRAMMING IN PASCAL is sure to hold the reader's interest

Stock No. #B23 — **Price** $14.95

UCSD Reference Book

- **A Reference Guide to the Complete UCSD PASCAL System**
- **Includes Information on Compiler Basic, Assembler and Editor**
- **Lists Actual P-Machine Codes**

This reference book can be a valuable and time-saving guide to thorough information on the UCSD PASCAL system. The easy-to-read manual provides fast access to pertinent data

Stock No. #B26 — **Price** $25.00

Reader Service—see page 178

equivalent of compiler declarations for these new words. The word CONSTANT (Table 8) takes an argument from the top of the stack and reads the next word from the console and makes that word the name of a dictionary entry which points to a type CONSTANT function whose value is the number on the stack.

The word variable works in exactly the same way but compiles its entry as type VARIABLE. The number on the stack is the initial value of the VARIABLE.

Notice that the words CONSTANT and VARIABLE are not part of the assembly-language kernel; they are definitions typed into the compiler. Although they could just as easily have been put into the kernel, and it was necessary to define extra constants in the kernel (CONS and VARBL) that give the addresses of the CONSTANT and VARIABLE type code, I think they give an interesting illustration of what the compiler is capable of doing along these lines.

That's the whole compiler. Table 9 contains a few more examples of function definitions using CONSTANTS and VARIABLES, including some contrasting LOOPing and recursive definitions. Listing 11 contains all the loose end routines not defined earlier. Table 10 contains a glossary of all the words defined in the kernel. This is useful to maintain as you add words and write new definitions, since it helps to keep you from rewriting functions which already exist and makes it easier to use them as well.

Extensions

I kept the code presented in these two articles as simple as possible to avoid complicating the explanations; as a result, it is far from perfect. Although it does run as described, it is lacking in a few respects, most notably error checking. Overstacking or overpopping the stacks will result in catastrophic failure of the system and unpredictable occurrences. If you make extensive use of this code, you will undoubtedly want to add as much of this kind of checking as you can. Additional necessary improvements include interfacing the code to your operating system or mass storage devices, so that you can store definitions on tape or floppy disk and load them when you want them.

Another useful feature is the ability to add comments to your definitions. FORTH defines the compiler word

```
- 0 VARIABLE TEMP                   define a temporary variable
0
- : *                               define a multiplication operator
    DUP TEMP POKEW                    - save the multiplicand
    SWAP 2 FOR                        - repeat the loop one less than the value of
                                        the multiplier times
              TEMP PEEKW +            - just add the multiplicand to the running
                                        total
          LOOP ;                    NOTE: This is a very inefficient algorithm and is
                                    here just for fun
0
- 10 10 *                           test *
100
- 3 4 *
12
- R                                 clear the stack
0
- : FAC                             an iterative definition of factorial
    DUP EQZ IF
          POP 1                     if argument is zero then value is 1
      ELSE
          1 SWAP 1 FOR              else it is the product of all numbers
                  I *
                LOOP                between N and 1
      THEN ;
0
- 6 FAC                             test FAC
720
- 0 FAC
1
- : FAC                             factorial can be defined recursively
    DUP EQZ IF
          POP 1                     if argument is zero then value is 1
      ELSE                          else call
          DUP DEC FAC               FAC with one less than the argument
          *                         and multiply result with argument
      THEN ;
1
- 3 FAC
6
```

Table 9.

Listing 8. FOR LOOP auxiliary definitions.

```
                ; FOR LOOP Auxilary Definitions

                ; FORINIT - initialize a FOR LOOP
                ; ENTRY: TOP   - starting value of loop index
                ;        TOP-1 - final value for loop index
                ; EXIT:  No Values Returned
                ; DESCRIPTION: pushes the final value plus one
                ; and the starting value minus one onto the loop stack.

06DB 4702       FORINIT DW      TCALL   ;threaded code because not critical
06DD DD026E023F         DW      SWAP,INC,LPUSH  ;final value
06E3 76023F07           DW          DEC,LPUSH  ;starting value
06E7 5B02               DW      TRET


                ; FORTEST - increment and test FOR LOOP index variable
                ; ENTRY: No Values Expected
                ; EXIT:  TOP - -1 if index variable has exceeded final value
                ;        TOP -  0 if index variable is still below final value
                ; DESCRIPTION:  increments index variable (found on top of
                ; loop stack) and compares it with final value plus one (next
                ; number on loop stack).  If equal, if leaves the values off
                ; of the loop stack and returns a -1 indicating termination of
                ; the loop otherwise it puts the values back on the loop stack
                ; and returns a zero.

06E9 EB06       FORTEST DW      $+2             ;machine code for efficiency
06EB CD4807             CALL    LPOPX   ;get index variable
06EE 13                 INX     D       ;and increment it
06EF D5                 PUSH    D       ;get final value
06F0 CD4807             CALL    LPOPX   ;DE <- final value
06F3 E1                 POP     H       ;HL <- index variable
06F4 E5                 PUSH    H
06F5 CD9102             CALL    MINUSH  ;subtract
06F8 19                 DAD     D
```

More

Listing 8 continued.

```
06F9 7C                 MOV     A,H
06FA B5                 ORA     L
06FB CA0C07             JZ      ENDFOR  ;if equal

06FE CD3407             CALL    LPUSHX  ;else put final value back
0701 D1                 POP     D       ;and put index variable back
0702 CD3407             CALL    LPUSHX
0705 210000             LXI     H,0     ;return a zero
0708 E5                 PUSH    H
0709 C33702             JMP     NEXT

070C E1         ENDFOR  POP     H       ;clear stack
070D 21FFFF             LXI     H,-1    ;return a minus one
0710 E5                 PUSH    H
0711 C33702             JMP     NEXT
                        PAGE
```

"(" (a left parenthesis) to be "read and ignore everything until the next right parenthesis," which permits comments to appear anywhere. Another method is to choose a single character, for example, a percent sign, and define it to be READLINE. This causes everything after it on the line to be skipped over.

Other extensions include adding a third parameter to the FOR LOOP to permit increments of the index variable other than one. You can do this directly or rewrite the code associated with FOR and make the increment an argument of the word LOOP (renamed to be distinct from the LOOP used with REPEAT). Words for performing more extensive reordering of the top three or four words on the stack are very useful (the use of the variable in the definition of the multiply operation in Table 9 would not have been necessary with a few of these). More extensive primitives for output are not hard to add. There is really no end to what can be done with this type of code. ■

:	the compiler
+	add top to top-1
−	subtract top from top-1
/MOD	divide top-1 by top, leave quotient at top-1 and remainder at top
INC	add one to top
DEC	subtract one from top
MINUS	negate top
EQ	if top = top-1, replace it with −1, else replace it with 0
EQZ	if top = 0, replace it with −1, else replace it with 0
I	push innermost FOR loop index
J	push second FOR loop index
K	push third FOR loop index
CLEAR	pop all elements off of the stack
POP	pop the top element off of the stack
SWAP	swap top and top-1
DUP	duplicate the top element of the stack leaving it at both top and top-1
READLINE	read a line from the console into the console buffer
SCAN	extract a word from the console buffer, leave a pointer to it at top-1, with −1 at top, or just 0 if end of line
PEEKB	replace top with byte located at (top)
PEEKW	replace top with word located at (top)
POKEB	replace byte at (top) with top-1; drop top and top-1
POKEW	replace word at (top) with top-1; drop top and top-1
PRINTS	print the string pointed to by top
CONBXA	convert binary number at top to ASCII string; leave a pointer to the string
CONAXB	try to convert ASCII string pointed to by top to binary number; leave value and −1 if successful, 0 if not
MATCH	compare two strings pointed to by top and top-1; leave string pointers plus true if same, false if not
LOOKUP	lookup string in dictionary, leave pointer to entry and true if found, else pointer to string and false
ENTER	enter string in dictionary; causes error if dictionary overflows
CMPLW	enter 16-bit word into code section of dictionary; causes error if dictionary overflows
MVBYTES	move a block of (top) bytes from (top-2) to (top-1)
EXECUTE	execute routine located at (top)
MEMORY	variable, value = top of available memory
CONS	constant, value = address of CONSTANT code
VARBL	constant, value = address of VARIABLE code

Table 10. Glossary of interpreter words.

References

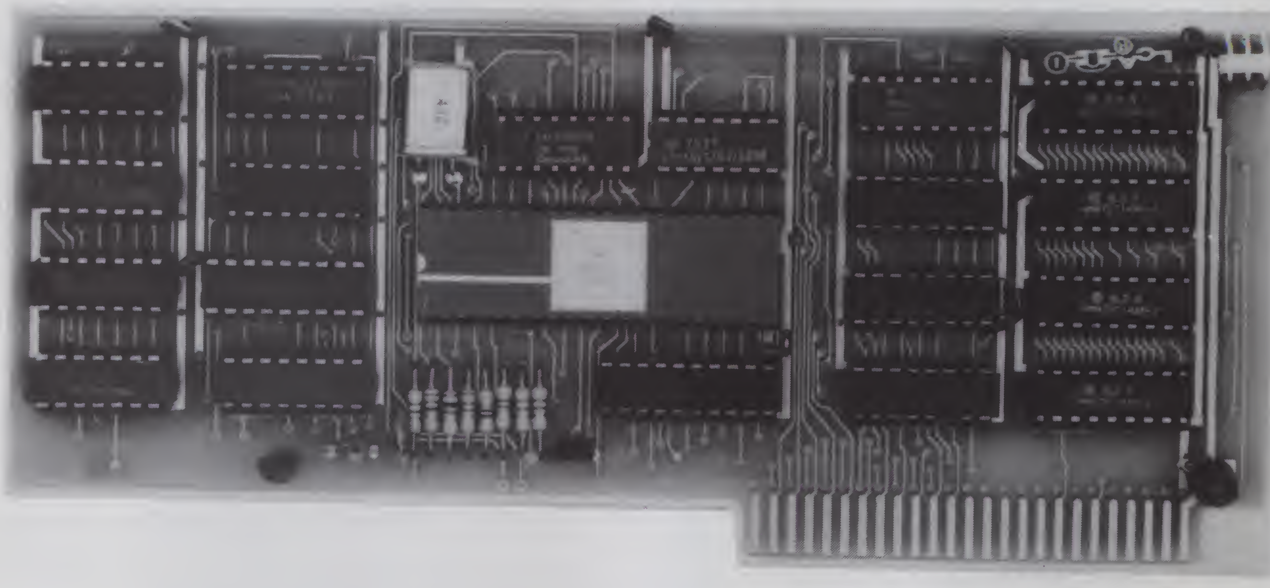
As mentioned in the beginning, this language is not FORTH. If you are interested in the real FORTH, or if you don't want to start your system from scratch, here are some useful sources:

● The FORTH Interest Group, PO Box 1105, San Carlos, CA 94070. These people sell a variety of FORTH literature and products including a source listing of a FORTH interpreter (for six popular CPUs) and an installation manual. They also publish a newsletter.

● "FORTH for Microcomputers," by John James, in *Dr. Dobb's Journal of Computer Calisthenics and Orthodontia*. Vol. 3, Issue 5. An introduction to standard FORTH including an explanation of how FORTH is usually interfaced to disks and CRT screens. A reprint is available from the FORTH Interest Group for $2.

FOR APPLE II AND APPLE II PLUS COMPUTERS

DoubleVision™



80 x 24 Video Display with Upper and Lower Case

COLUMNS LINES

• is a hardware board that may be plugged into any slot in Apple II or Apple II Plus 32K or 48K Disks • full 128 ASCII character set, including control characters • fully programmable cursor • built in light pen capability • inverse video • full cursor control • works with 50/60Hz • has 2k of its own screen memory • has its own video output jack that must be connected to a monitor (or a high band width black & white TV thru a good RF modulator). Color TV's produce a poor display and are not recommended. • permits you to connect another monitor (or a T.V. set thru RFmod) to the Apple video output jack • displays 24 lines of 80 column text — programmable for different values • permits you to have graphics on Apple video output • video output and Apple video output may be connected to one monitor thru optional video switch • is active only when addressed for reading from or writing to • accepts lower case input from keyboard by use of escape key. (no modification required) or direct use of shift key (1-wire connection from shift key pad to DoubleVision required). • is compatible with the latest version of various word processing software packages. Presently these include Apple-pie 2.0— Programma International, Easywriter Professional system—Informational Unlimited, Text Editor/Formatter—Peripheral's Unltd. (when ordering from these companies, please ask for versions compatible with DoubleVision). All software available from Computer Stop when released. • Peripheral's Unltd. B.I.T.S. and P.I.T.S. and Southeastern Software's "DATA CAPTURE" with Micromodem and communication card. These packages give ability to upload, transfer and download files from remote computers, and all at 80 columns! • Programma Int. latest assembler LISA V:20 will support full 80 column display • is transparent for use with Basic and Pascal • software on disk for easy modification and adaptation for different applications • completely commented source listing of software and hardware schematics available • PASCAL (optional) • becomes the console when installed in Pascal • Permits 80 column text processing with full upper/lower case while using Pascal's editor • must be plugged into slot 3 when operating with Pascal

Available now at your local computer store....... $295.00

Call Computer Stop for Store nearest you

Calif. Residents add 6% Sales Tax

Shipping, Insurance, Handling, extra

*Apple is a Registered TM of Apple Computers, Inc.

Dealer inquiries invited.
Contact:

COMPUTER STOP CORP.
2545 West 237th St.
Suite L
Torrance, CA 90505
539-7671

The Computer Stop
16919 Hawthorne Blvd.
Lawndale, CA 90260 ✓283
(213) 371-4010

MON. - SAT.
10-6

```
; FOR - compile beginning of FOR loop
; EXIT: TOP   - address of middle of conditional jump
;               to be filled in by LOOP
;       TOP-1 - address of FORTEST instruction, the
;               beginning of the FOR loop
; DESCRIPTION: Compiles FORINIT and FORTEST insttructions.
; Follows the FORTEST with a jump-if-not-zero. Leaves the
; address of the FORTEST instruction and the address of
; the second word of the conditional jump instruction
; on the stack to be used by LOOP.  Notice that the stack
; is left in the same condition a REPEAT and UNTIL would
; have left it in.

067F 4702         FOR    DW     TCALL           ;threaded code
0681 C702DB06B3          DW     TPUSH,FORINIT,CMPLW     ;compile FORINIT
0687 6404AB0276          DW     CODE,PEEKW,DEC  ;save next address
068D C702E906B3          DW     TPUSH,FORTEST,CMPLW     ;compile FORTEST
0693 C7021803B3          DW     TPUSH,IFNZ,CMPLW        ;compile conditional jump
0699 6404AB02            DW     CODE,PEEKW      ;save next address
069D 6C03B305            DW     ZERO,CMPLW      ;compile a zero temprarily
06A1 5B02                DW     TRET
```

Listing 9. FOR—compile beginning of FOR loop.

```
; FOR LOOP Index Variables
;       The following functions implement the words I, J and K  wh
; lookup the current value of the index variable for the innermost
; second and outermost FOR LOOPs.

; I, J , K - FOR LOOP index Variables
; ENTRY: No Values Expected
; EXIT:  TOP - value of index variable
; DESCRIPTION: performs a table lookup on the Loop Stack.
; 'I' selects the first element on the stack, 'J' selects
; the third and 'K' selects the fifth.  (Every other
; element contains the terminating value of the loop variable.)

0714 1607      XI     DW     $+2      ;machine code
0716 11FFFF           LXI    D,-1     ;1st element for I
0719 C32907           JMP    XN

071C 1E07      XJ     DW     $+2      ;machine code
071E 11FDFF           LXI    D,-3     ;3rd element for J
0721 C32907           JMP    XN

0724 2607      XK     DW     $+2      ;machine code
0726 11FBFF           LXI    D,-5     ;5th element for K

0729 2A5307    XN     LHLD   LSTACK   ;get loop stack pointer
072C 19               DAD    D        ;offset to desired element
072D 56               MOV    D,M      ;DE <- element
072E 2B               DCX    H
072F 5E               MOV    E,M
0730 D5               PUSH   D        ;stack the value
0731 C33702           JMP    NEXT
                      PAGE
```

Listing 10. FOR LOOP index variables.

Listing 11a. New additions to compiler version.

```
             ; New Additions to Compiler Version

             ; IFGT - jump if TOP-1 > TOP
             ;
             ; Version 1.5 new function

0333 3503    IFGT   DW     $+2      ;CODE
0335 D1             POP    D        ;DE <- TOP
0336 E1             POP    H        ;HL <- TOP-1
0337 CD4003         CALL   SUB16    ;HL <- TOP-1 - TOP
033A D2F702         JNC    JUMP1    ;jump if TOP is smaller
033D C30D03         JMP    SKIP     ;otherwise don't

             ; SUB16 - 16 bit subtraction
             ; ENTRY: HL - subtrahounihum
             ;        DE - the other one
             ; EXIT:  HL - la difference (HL-DE)
             ;        Carry set if HL<DE
             ; REGISTER USE: A PSW

0340 7D      SUB16  MOV    A,L
0341 93             SUB    E
0342 6F             MOV    L,A
0343 7C             MOV    A,H
0344 9A             SBB    D
0345 67             MOV    H,A
0346 C9             RET

      ; CFLAG - local variable for compiler
      ; DESCRIPTION: set to zero at initiation of compile loop,
```

More

Listing 11a continued.

```
                ; set to -1 by semicolon and error routines to signal the
                ; end of compilation.

0347 7C03       CFLAG   DW      VARIABLE
0349 0000               DW      0

034B 6303       CONADD  DW      CONSTANT        ;address of CONSTANT code
034D 6303               DW      CONSTANT
034F 6303       VARADD  DW      CONSTANT        ;address of VARIABLE code
0351 7C03               DW      VARIABLE

                ; GETWORD - get a word from input
                ; ENTRY: No Parameters expected
                ; EXIT:  TOP   - pointer to a word read from console
                ; DESCRIPTION: Uses SCAN to retrieve next word on
                ; line.  If there is no next word, reads a line
                ; from the console and uses SCAN again.

0353 4702       GETWORD DW      TCALL           ;threaded code

0355 E407       GW1     DW      SCAN            ;get next word
0357 18036003           DW      IFNZ,GW2-1      ;if gotten
035B 9904               DW      READLINE        ;else read a line
035D F5025403           DW      JUMP,GW1-1      ;and try again
0361 5B02       GW2     DW      TRET
                        PAGE
```

```
        ; Stack Tests
        ;
        ; These auxiliary functions test the stack for conditions
        ; similar to those tested by the conditional jumps.  They leave
        ; the results of their tests on the stack (True = -1, False = 0)
        ; for testing by IF or UNTIL.  Tests like these replace the conditional
        ; jumps in the compiler language which has no GOTO statements.
        ;       The functions are written in machine language for efficiency
        ; since they tend to be executed frequently.

        ; EQ - test if TOP = TOP-1
        ; DESCRIPTION: Pops they top two elements of the stack and replaces
        ; them with a -1 if they were equal or a zero if they were not.

06B3 B506       EQX     DW      $+2             ;CODE
06B5 E1                 POP     H               ;get TOP
06B6 D1                 POP     D               ;get TOP-1
06B7 CD4003             CALL    SUB16           ;subtract
06BA 7C                 MOV     A,H             ;test for zero
06BB B5                 ORA     L
06BC 210000             LXI     H,0             ;anticipate false
06BF C2C506             JNZ     EQX1            ;if not equal
06C2 21FFFF             LXI     H,-1            ;else return true
06C5 E5         EQX1    PUSH    H               ;stack the answer
06C6 C33702             JMP     NEXT            ;and continue

                ; EQZ - test if TOP = 0
                ; DESCRIPTION: Pops the top of the stack and replaces it with
                ; a -1 if it was zero or a zero if it was not.


06C9 CB06       EQZ     DW      $+2             ;CODE
06CB E1                 POP     H               ;get TOP
06CC 7C                 MOV     A,H             ;test for zero
06CD B5                 ORA     L
06CE 210000             LXI     H,0             ;anticipate false
06D1 C2D706             JNZ     EQZ1            ;if not zero
06D4 21FFFF             LXI     H,-1            ;else return true
06D7 E5         EQZ1    PUSH    H               ;stack the answer
06D8 C33702             JMP     NEXT
                        PAGE
```

Listing 11b. Stack tests.

Listing 11c. CMPLW—compile a word of code.

```
                ; CMPLW - compile a word of code
                ; ENTRY: TOP   - 16 bit word to be compiled
                ; EXIT:  No Values Returned
                ; DESCRIPTION:  Stores the word in the code section
                ; of the dictionary.  Updates the CODE pointer.
                ; Yields an error if the dictionary is full.

05B3 4702       CMPLW   DW      TCALL              ;threaded code
05B5 6404AB02E5         DW      CODE,PEEKW,DUP,INC,INC ;add two to code
05BF E5026004AB         DW      DUP,NAMES,PEEKW ;compare with names
05C5 3303D005           DW      IFGT,CERROR-1   ;if memory full
05C9 6404BD02           DW      CODE,POKEW      ;else store new code pointer
05CD BD02               DW      POKEW           ;and compile word
05CF 5B02               DW      TRET

05D1 C702A205   CERROR  DW      TPUSH,EEMSG     ;print bad new
05D5 EB05               DW      UERROR
```

More

Listing 11c continued.

```
05D7 5B02                DW      TRET


               ; ERROR - error during compilation
               ; ENTRY:  TOP   - pointer to error message
               ; DESCRIPTION:  Prints message and aborts compilation

05D9 4702       ERROR    DW      TCALL                    ;threaded code
05DB A307                DW      PRINTS                   ;print message
05DD 74034703BD          DW      NEGONE,CFLAG,POKEW       ;abort compilation
05E3 7403F504B5          DW      NEGONE,CONBUF,POKEB      ;skip remainder of input l
05E9 5B02                DW      TRET

               ; UERROR - 'UNDO' ERROR
               ; DESCRIPTION: same as ERROR but first 'undoes' the
               ; partially completed compilation.

05EB 4702       UERROR   DW      TCALL                    ;threaded code
                                                          ;reset NAMES pointer
05ED 6004AB02E5          DW      NAMES,PEEKW,DUP ;current names pointer
05F3 A1027E02            DW      PEEKB,TADD      ;plus length of string
05F7 6E02E5026E          DW      INC,DUP,INC,INC ;plus three(save ptr to old CODE)
05FF 6004BD02            DW      NAMES,POKEW     ;gives new names pointer
                                                          ;reset CODE pointer
0603 AB02                DW      PEEKW           ;code pointer for aborted entry
0605 6404BD02            DW      CODE,POKEW      ;replace current code pointer
0609 D905                DW      ERROR                    ;normal error
060B 5B02                DW      TRET
                         PAGE
```

```
                ; System Variables

                ; Names - address of interpreter dictionary names

0460 7C03       NAMES    DW      VARIABLE        ;threaded code variable
0462 0909                DW      NAMEBEG         ;beginning of names

                ; CODE - address of the dictionary code section

0464 7C03       CODE     DW      VARIABLE
0466 0909                DW      NAMEBEG         ;assembles just before names

                ; CDICT - address of compiler dictionary names

0468 7C03       CDICT    DW      VARIABLE
046A 0C0A                DW      CNAMES
                         PAGE
```

Listing 11d. System variables.

```
                 ; MVBYTES - nove a block of bytes forward
                 ; ENTRY: TOP   - byte count
                 ;        TOP-1 - target address
                 ;        TOP-2 - source address
                 ; EXIT:  No Values Returned
                 ; DESCRIPTION: Moves the block of bytes beginnin
                 ; at the source address to the target address.

08F7 F908        MVBYTES DW      $+2     ;CODE subroutine
08F9 C1                  POP     B       ;BC <- count
08FA D1                  POP     D       ;DE <- target
08FB E1                  POP     H       ;HL <- source

08FC 7E          MVB1    MOV     A,M     ;get next byte
08FD 12                  STAX    D       ;store it
08FE 23                  INX     H       ;bump source pointer
08FF 13                  INX     D       ;bump target pointer
0900 0B                  DCX     B       ;decrement count
0901 78                  MOV     A,B     ;test for zero
0902 B1                  ORA     C
0903 C2FC08              JNZ     MVB1
0906 C33702              JMP     NEXT    ;exit when finished
                         PAGE
```

Listing 11e. MVBYTES—move a block of bytes forward.

Listing 11f. The names in the dictionary.

```
; The Names in the dictionary
; Notice that the actual printed names are chosen for typing
; convenience and do not necessarily match the internal names
; which must conform to the assembler's rules. Also, not all
```

More

Listing 11f continued.

```
                ; functions have been included here.

0909 =          NAMEBEG EQU     $           ;BEGINNING OF INTERPRETER DICTIONARY
0909 013A         DB      1,':'             ;note that the compiler is a word in the
090B C303         DW              COMPILE   ;interpreter's  dictionary

090D 012B         DB      1,'+'             ;arithmetic words
090F 7E02         DW              TADD
0911 012D         DB      1,'-'
0913 9902         DW              TSUB
0915 042F4D4F44   DB      4,'/MOD'          ;(not '/'; yields quotient and remainder)
091A BC08         DW              DIV
091C 03494E43     DB      3,'INC'
0920 6E02         DW              INC
0922 03444543     DB      3,'DEC'
0926 7602         DW              DEC
0928 054D494E55   DB      5,'MINUS'
092E 8702         DW              MINUS
0930 024551       DB      2,'EQ'            ;stack tests
0933 B306         DW              EQX
0935 0345515A     DB      3,'EQZ'
0939 C906         DW              EQZ
093B 0149         DB      1,'I'             ;FOR LOOP index values
093D 1407         DW              XI
093F 014A         DB      1,'J'
0941 1C07         DW              XJ
0943 014B         DB      1,'K'
0945 2407         DW              XK
0947 05434C4541   DB      5,'CLEAR'         ;stack manipiulating words
094D ED02         DW              CLEAR
094F 03504F50     DB      3,'POP'
0953 D702         DW              TPOP
0955 0453574150   DB      4,'SWAP'
095A DD02         DW              SWAP
095C 03445550     DB      3,'DUP'
0960 E502         DW              DUP
0962 0546495253   DB      5,'FIRST'
0968 C107         DW              FIRST
096A 055045454B   DB      5,'PEEKW'
0970 AB02         DW              PEEKW
0972 055045454B   DB      5,'PEEKB'
0978 A102         DW              PEEKB
097A 05504F4B45   DB      5,'POKEW'
0980 BD02         DW              POKEW
0982 05504F4B45   DB      5,'POKEB'
0988 B502         DW              POKEB
                  PAGE
098A 0852454144   DB      8,'READLINE'      ;soe input/output words
0993 9904         DW              READLINE
0995 045343414E   DB      4,'SCAN'
099A E407         DW              SCAN
099C 065052494E   DB      6,'PRINTS'
09A3 A307         DW              PRINTS
09A5 06434F4E41   DB      6,'CONAXB'
09AC 8508         DW              CONAXB
09AE 06434F4E42   DB      6,'CONBXA'
09B5 3108         DW              CONBXA
09B7 054D415443   DB      5,'MATCH'         ;miscellaneous
09BD 6C04         DW              MATCH
09BF 064C4F4F4B   DB      6,'LOOKUP'
09C6 2E04         DW              LOOKUP
09C8 05454E5445   DB      5,'ENTER'         ;enter word in dictionary
09CE 5C05         DW              ENTER
09D0 05434D504C   DB      5,'CMPLW'         ;compile a word of code
09D6 B305         DW              CMPLW
09D8 0445584543   DB      4,'EXEC'
09DD 1901         DW              EXEC
09DF 074D564259   DB      7,'MVBYTES'
09E7 F708         DW              MVBYTES
09E9 0745584543   DB      7,'EXECUTE'
09F1 9104         DW              EXECUTE
09F3 064D454D4F   DB      6,'MEMORY'        ;VARIABLE; value=top of memory
09FA 7803         DW              MEMORY
09FC 04434F4E53   DB      4,'CONS'          ;CONSTANT;value=address of CONSTANT code
0A01 4B03         DW              CONADD
0A03 0556415242   DB      5,'VARBL'         ;CONSTANT;value=address of VARIABLE code
0A09 4F03         DW              VARADD
0A0B 00           DB      0                 ;END OF INTERPRETER DICTIONARY

                                            ;BEGINNING OF COMPILER DICTIONARY
0A0C =          CNAMES  EQU     $
0A0C 024946       DB      2,'IF'
0A0F 0D06         DW              IFX
0A11 045448454E   DB      4,'THEN'
0A16 1F06         DW              THEN
0A18 04454C5345   DB      4,'ELSE'
0A1D 2D06         DW              ELSEX
0A1F 0652455045   DB      6,'REPEAT'
0A26 4B06         DW              REPEAT
0A28 05554E5449   DB      5,'UNTIL'
0A2E 5506         DW              UNTIL
0A30 044C4F4F50   DB      4,'LOOP'
0A35 6706         DW              LOOP
0A37 03464F52     DB      3,'FOR'
0A3B 7F06         DW              FOR
0A3D 013B         DB      1,';'
0A3F A306         DW              SEMI
0A41 00           DB      0                 ;END OF COMPILE DICTIONARY

0A41 =          NAMEEND EQU     $-1
0139 =          DICSIZE EQU     NAMEEND-NAMEBEG-1    ;dicsize in bytes
0130 =          IDICSIZ EQU     CNAMES-NAMEBEG       ;inter diesize in bytes
```


✓ Reader Service—see page 178

```
                        ; Initialization Code
                        ; Executed at start up of sytem  but eventually overwritten
                        ; by the expanding dictionary

                        ; DICMOVE - move the dictionary names to the top of available memo

0A42 2A7A03    DICMOVE LHLD     MEMORY+2         ;DE <- top of memory
0A45 EB                XCHG
0A46 21410A            LXI      H,NAMEEND        ;HL <-source
0A49 013901            LXI      B,DICSIZE        ;BC <- byte count

                                                 ;transfer loop
0A4C 7E        DIC1    MOV      A,M              ;get next byte
0A4D 12                STAX     D                ;move it
0A4E 2B                DCX      H                ;dec source pointer
0A4F 1B                DCX      D                ;dec target pointer
0A50 0B                DCX      B                ;dec count
0A51 78                MOV      A,B              ;test for zero
0A52 B1                ORA      C
0A53 C24C0A            JNZ      DIC1

0A56 EB                XCHG                      ;set dictionary variableS
0A57 23                INX      H
0A58 226204            SHLD     NAMES+2
0A5B 110301            LXI      D,IDICSIZ
0A5E 19                DAD      D
0A5F 226A04            SHLD     CDICT+2
0A62 C9                RET
0A63                   END
```

Listing 11g. Initialization code.

The author's not name-calling; he's just describing his ultra-economical data communications connection.

Cheap, Dumb Apple

By Bill Hubbard

Have you ever wanted to use your Apple as a dumb terminal? This article describes an inexpensive way to do it.

Normally, you'd need a modem costing about $200 and an interface board for another $200. But the Electronic Systems Modem (part 109) costs $27.50. Two recent *Microcomputing* articles describe how to do it (January 1980, p. 52, and February 1980, p. 168).

Now that the high cost of modems has been circumvented, what about an interface? I used the Apple's game paddle connector. You can drive a printer or Teletype from there.

The simple serial output section in the *Apple Red Book* has details about this. But it doesn't tell you how to come back in. The game paddle connector is a two-way street, so if you had a driver program to read one of the game switches as serial ASCII, you could display it on the screen. See listings.

The only additional hardware you'll need is a 16-pin DIP carrier, a ¼ A mini-fuse and two 100 ohm resistors. A minor enhancement is a 3k resistor and an LED to give a visual indication of the incoming signal. See Fig. 1.

This version uses TTL signals both from the Apple's AN0 output and into its SW2 input. It uses the Apple's 5-volt power supply. I used a slightly different phone connection. Results have been good. See Fig. 2.

Correspondence concerning this article should be addressed to Bill Hubbard, 2204 Manlyn Road, Richmond, VA 23229.

To use it, just enter the program by hand, cassette or disk, and from monitor mode type 300 G. All subsequent keyboard characters will modulate the phone line and be displayed. Incoming characters should show up as plain English on your screen. The cursor and all monitor functions are disabled until you hit the reset key.

The driver program resides at locations 300 through 35C and can easily be keyed in by hand and saved to disk or cassette. It consists of an idle loop to continuously check keyboard and input switch, an input routine and an output routine. Miniassembler and hex listings are included.

This program works only at 110

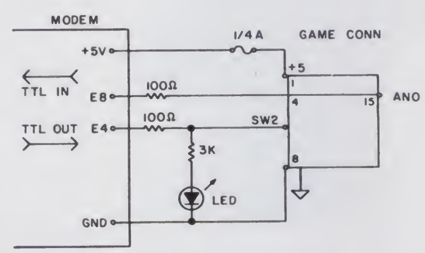


Fig. 1. Connections from inexpensive modem to the Apple game paddle connector allowing it to be used as a timeshare terminal.

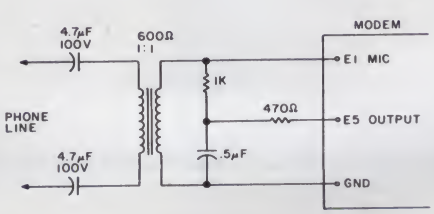


Fig. 2. Simple method of connecting modem directly to phone line.

300	LDA	C000	Keybd
303	BPL	030B	
305	JMP	0339	Output
308	STA	C010	Clear
30B	LDA	C063	Ck Sw
30E	BMI	0300	Loop
310	LDY	#$08	Bits
312	LDA	#$25	½ bit
314	JSR	FCA8	Wait
317	LDA	C063	Re Ck Sw
31A	BMI	0300	False?
31C	PHA		Save Char
31D	LDA	#$3B	1 bit
31F	JSR	FCA8	Wait
322	LDA	C063	Ck Sw
325	BPL	032B	Skip
327	SEC		Mark
328	JMP	032C	Skip
32B	CLC		Space
32C	PLA		Pull Char
32D	ROR		Rotate
32E	DEY		Dec Bit
32F	BNE	031C	Done?
331	ORA	#$80	Hi Bit
333	JSR	FDF0	Display
336	JMP	0300	Loop
339	STA	03FF	Save Char
33C	LDY	#$0B	11 Bits
33E	CLC		
33F	PHA		Push Char
340	BCS	0347	Skip
342	STA	C058	Space
345	BCC	034A	Skip
347	STA	C059	Mark
34A	LDA	#$3A	1 bit
34C	JSR	FCA8	Wait
34F	PLA		Pull Char
350	ROR		Rotate
351	DEY		Dec Bit
352	BNE	033F	Done?
354	STA	C010	Clear
357	LDA	03FF	Get Char
35A	JMP	0331	Go Display

Assembly listing of dumb terminal driver.

300	AD	00	C0	10	06	4C	39	03
308	8D	10	C0	AD	63	C0	30	F0
310	A0	08	A9	25	20	A8	FC	AD
318	63	C0	30	E4	48	A9	3B	20
320	A8	FC	AD	63	C0	10	04	38
328	4C	2C	03	18	68	6A	88	D0
330	EB	09	80	20	F0	FD	4C	00
338	03	8D	FF	03	A0	0B	18	48
340	B0	05	8D	58	C0	90	03	8D
348	59	C0	A9	3A	20	A8	FC	68
350	6A	88	D0	EB	8D	10	C0	AD
358	FF	03	4C	31	03	00	00	00

Hex listing of dumb terminal driver.

baud, only in uppercase ASCII, only in half duplex and only in originate mode. After all, it's only a cheap, dumb Apple.■

SUPER BRAIN QD™

Once in a great while someone comes along with a simple improvement for an already great product. Take our SuperBrain, for example. Really a simple concept. A high-powered, low cost microcomputer packaged in an attractive desk top cabinet. So how do you improve on that?

WE DID IT...

It wasn't enough that our SuperBrain had such standard features as twin double density 5¼" drives with over 300,000 bytes of disk storage. A full 32K of dynamic RAM - expandable to 64K in seconds. A CP/M* Disk Operating System which assures compatibility to literally hundreds of application packages presently available. A crisp, 12" non-glare screen with a full 24 line by 80 column display. A full ASCII keyboard with a separate keypad and individual cursor control keys. Twin RS232C serial ports for fast and easy connection to a modem and/or a printer. And, dual Z80 processors which operate at 4 MHZ to insure lightning-fast program execution. No, it wasn't enough. So we made it better.

ANNOUNCING SUPERBRAIN QD...

Our new QD model has all of the features of our phenomenally popular SuperBrain with the addition of double-sided disk drives and an extra 32K of dynamic RAM. So, for only a modest increase in price, you can order your next SuperBrain with more than twice the disk and memory storage capability. But, best of all, the new QD model has the same tough, rugged construction and exceptional quality that made our SuperBrain such a success.

HOW DID WE DO IT?

The secret of SuperBrain QD's incredible disk storage lies within our new double-density double-sided disk drives. A total of nearly 720,000 bytes of data are formatted on two specially designed 5¼" drives. And that's more than enough to get you started with most serious small business applications. And SuperBrain QD's standard 64K of dynamic RAM will handle even the most complicated programming tasks.

Of course, if you're into megabytes instead of kilobytes, you may think neither SuperBrain is right for you. Not so! Intertec offers 20-96 megabytes of hard-disk storage which connects in seconds to either the SuperBrain or SuperBrain QD. So, your original investment is always protected. As you grow. No matter how much your needs expand.

BUT IS IT RELIABLE?

Our best salesmen are our present users. Not only have SuperBrain users been impressed with the inherent reliability of the system, they tell us that no other microcomputer system available today offers such a unique modular design concept. Just about the only tool required to easily maintain the system is a common screwdriver. And Intertec's total commitment to product service and customer support, with service outlets in most major cities, insures your original investment will be a valuable one for many years to come.

THE DECISION IS YOURS.

Whether your next SuperBrain is a regular model or our QD version, you will have the satisfaction of knowing you purchased what is becoming one of the world's most popular microcomputer systems. And regardless of which model you choose, you'll probably never outgrow it because you can keep expanding it.

So, call or write us today for more information. Intertec systems are distributed worldwide and may be available in your area now.

INTERTEC DATA SYSTEMS®

2300 Broad River Rd., Columbia, SC 29210
(803) 798-9100 TWX: 810-666-2115 ✓343

*Registered trademark of Digital Research, Inc.

grams that are begun by executing a text file are rare in my collection, I did not feel it worth the trouble to add this ability. However, it should be fairly simple to do. The logic in line 100 should be changed to check for a "B" or a "T." Then a new line must be added which goes to either existing line 110 or a new one which EXEC's A$. Also, line 60 must be modified to prevent the "text" warning from being printed.

Since I wrote Auto-menu, I have used the SCRN technique in several other programs. There are undoubtedly many applications for this ability to easily access the contents of the display screen. I would enjoy hearing from anyone who uses this method in other situations. ■

If you can type, you can handle your accounting, word processing and much more on ISC's Small Business Computer!

The Intecolor® 8963 is just one of a complete line of ISC desk-top computers designed for businessmen who don't know how—or don't have time—to write programs. It's CP/M® compatible, so you can choose from hundreds of CP/M business programs—programs that have been proven in hundreds of actual applications.

Programs like General Ledger, Accounts Receivable and Payable, Payroll, Mailing List and Inventory Control are now available *in color.* You'll comprehend data faster, thanks to the proven readability of ISC's eight-color display.

To compose letters and other documents quickly and easily, you'll want ISC's unique *color-coded* Word Processing program. With an optional printer, you can print out as many mistake-free originals as you want!

The Intecolor 8963 gives you all the benefits of color graphics computing at a price that's competitive with black-and-white. It's complete with 19" color graphics display, typewriter-like keyboard, dual disk drive for data storage —even a color version of Microsoft® Business BASIC for those of you who *do* want to program.

Don't let your business get behind the times. Call your ISC sales representative or visit your nearest Factory Authorized ISC dealer and get a "hands on" demonstration *today.*

Color Communicates Better℠

* U S domestic price Unretouched photo of screen. CP M is a registered trademark of Digital Research Corp.

OEM Quantity Discounts are available to Qualified Dealers and Volume Users of 25 Systems (or greater) per year. Call your nearest ISC Representative listed below.

ISC SALES REPRESENTATIVES: AL: 205/883-8660. **AZ:** 602/994-5400. **AR:** (OK) USDATA Assoc. 918/252-9646. **CA:** Alhambra 213/281-2280. Goleta 805/964-8751. Irvine 714/557-4460. Los Angeles 213/476-1241. Los Altos 415/948-4563. San Diego 714/292-8525. **CO:** 303/355-2363. **CT:** 203/624-7800. **DE:** (PA) 215/542-9876. **DC:** (VA) 703/569-1502. **FL:** Ft. Lauderdale 305/776-4800. Melbourne 305/723-0766. Orlando 305/425-5505. Tallahassee 904/878-6642. **GA:** Atlanta 404/455-1035. **HI:** 808/524-8633. **ID:** (UT) 801/292-8145. **IL:** (No.) 312/564-5440. (So. MO) 816/765-3337. **IN:** (IL) 312/564-5440. **IA:** (Scott County Only) 312/564-5440. (MO) 816/765-3337. **KS:** (MO) 816/765-3337. **KY:** 606/273-3771. **LA:** 504/626-9701. **ME:** (MA) 617/729-5770. **MD:** (VA) 703/569-1502. **MA:** 617/729-5770. **MI:** Brighton 313/227-7067. Grand Rapids 616/363-9839. **MN:** 612/645-5816. **MS:** (AL) 205/883-8660. **MO:** 816/765-3337. **MT:** (CO) 303/355-2363. **NB:** (MO) 816/765-3337. **NH:** (MA) 617/729-5770. **NJ:** (No.) 201/224-6911. (So.) 215/542-9876. **NV:** (AZ) 602/994-5400. **NM:** 505/292-1212. **NY:** Metro/LI(NJ) 201/224-6911. N. Syracuse 315/699-2651. Fairport 716/223-4490. Utica 315/732-1801. **NC:** 919/683-1580. **ND:** (MN) 612/645-5816. **OH:** Cleveland 216/398-9200. Dayton 513/435-7684. **OK:** USDATA Assoc. 918/252-9646. **OR:** 503/644-5900. **PA:** (E) 215/542-9876. (W) 412/922-5110. **RI:** (MA) 617/729-5770. **SC:** 803/798-8070. **SD:** (MN) 612/645-5816. **TN:** 615/482-5761. **TX:** Austin 512/454-3579. Dallas 214/661-9633. El Paso Area (Las Cruces, NM) 505/524-9693. Houston Only 713/681-0200. **UT:** 801/292-8145. **VT:** (MA) 617/729-5770. **VA:** 703/569-1502. **WA:** 206/455-9180. **WV:** (PA) 412/922-5110. **WI:** (IL) 312/564-5440. **WY:** (CO) 303/355-2363.

EUROPEAN EXPORT SALES: EUROPE: (MA) 617/661-9424. **BELGIUM:** Noron S.A., Brussels 02-24236-04. **DENMARK:** 02-913255. **FRANCE:** Rueil Malmaison 749-47-65. Paris 33-1-306-4606. **GREECE:** Athens 642-1368. **ITALY:** Milano 02600733. **THE NETHERLANDS:** Poeldijk 01749-47640. Amsterdam 020-360904. **SPAIN:** Mecorsa, Barcelona. TELEX 813-53908. **SWEDEN:** Teleinstruments AB, Vallingby. 08-380-370. **SWITZERLAND:** Mutschellen 057-546-55. **UNITED KINGDOM:** Bournemouth 0201671181. **ICELAND:** Benco. Ltd., Reykjavik. 091-21945. **WEST GERMANY:** Koblenz 01149-31025 6. **AUSTRALIA & NEW ZEALAND:** Auckland 876-570. Canberra 58-1811. Chermside 59-6436. Christchurch 796-210. Melbourne 03-543-2077. Sydney 02-808-1444. Wainuiomata 644-585. **CANADA:** Dorval 514/636-9774. Ottawa 613/224-1391. Toronto 416/787-1208. Vancouver 604/684-8625. **CENTRAL AND SOUTH AMERICA & CARIBBEAN:** (GA) 404/394-9603. **MEXICO:** Monterrey 564-876. **FAR EAST:** (CA) 213/382-1107. **HONG KONG:** Comptec, Hong Kong. 03-463-9921. **JAPAN:** (Tokyo) (03) 463-9921. **TAIWAN:** (Taipei) 02 7022156. **MIDDLE EAST:** (GA) 404/581-0243. **EGYPT:** 809933. **KUWAIT:** Kuwait 438-180/1/2. **LEBANON:** Beirut 221731/260110. **SAUDI ARABIA:** Jeddah 27790. Riyadh 25083-39732

For sales and service in other countries contact ISC headquarters in Norcross, GA., U.S.A.

Intelligent Systems Corp.® ✓76

Intecolor Drive □ 225 Technology Park/Atlanta □ Norcross, GA 30092 □ Telephone 404/449-5961 □ TWX 810-766-1581

No formal introductions are necessary; just sit back and read how you can add color graphics to your computer.

Meet the Electric Crayon

By Michael B. Roberts

Looking for low-cost color graphics for your computer? The Electric Crayon by Percom could be just what you need.

This versatile unit offers ten software-selectable display modes ranging from an alphanumerics mode, which displays 16 lines of 32 characters via a built-in character generator, to a full graphics mode, which displays 192 rows of 256 points. Up to eight colors can be used, depending on the mode selected.

An on-board 6802 microprocessor executes a graphics control program contained in read-only memory, accepting commands from a keyboard or computer connected to one of the two dual I/O ports. Commands can be input to erase the screen, set display mode and color, plot points, draw horizontal and vertical lines and load machine-language programs or data into the Electric Crayon's memory.

Description

The Electric Crayon is housed in an attractive metal enclosure measuring 9 × 12 × 2.5 inches (23 × 30.5 × 6.4 cm). The circuitry is contained on a single printed circuit board mounted to the bottom of the cabinet. Two card-edge connectors for the I/O ports extend through slots in the rear of the cabinet. The rear panel also contains the power switch, fuse holder, power cord and an F61 coaxial connector for the video output.

The Electric Crayon exposed.

The heart of the unit is the Motorola MC6847 video display generator (VDG) and its friend, the MC1372 chroma and video modulator. The MC6847 reads picture information from the Crayon's on-board random access display memory and translates it into a video signal which is sent to the MC1372. Eight control inputs to the VDG provide mode and color selection under software control for the picture elements in the display. The MC6847 includes an internal character generator that can display 64 ASCII uppercase letters, numerals and symbols in two colors and inverse video (black characters on a colored background).

The MC1372 converts the display information from the VDG into a composite video signal, which is then fed to a monitor. As supplied, the video output from the Electric Crayon will drive a video input on a monitor or modified TV receiver. Instructions are furnished for easily converting the output to a modulated video carrier on UHF channels 33–40. A

Michael B. Roberts, 681 Overland Ave., Cincinnati, OH 45226.

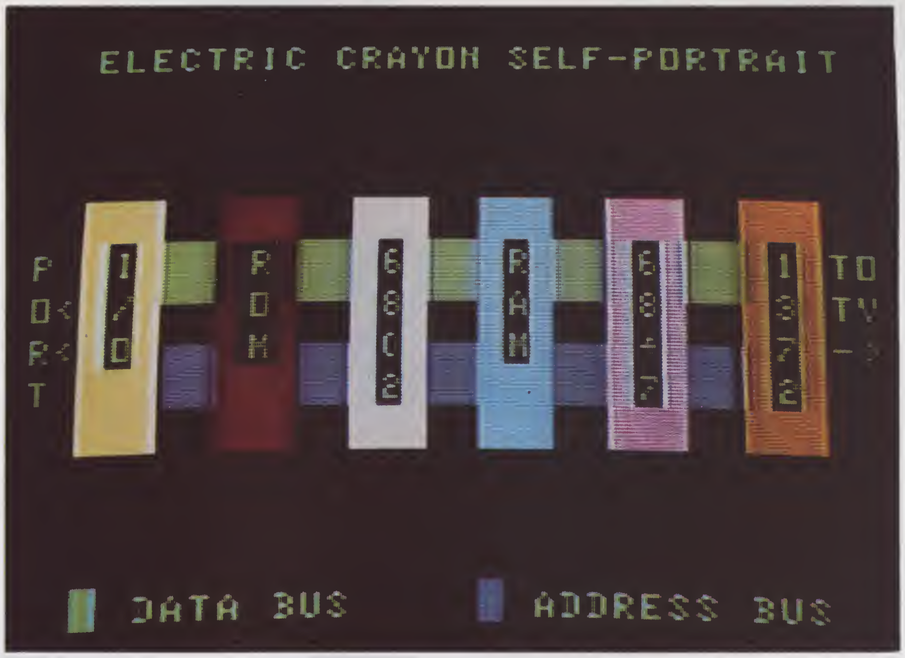
3.579545 MHz crystal controls an internal clock; an output on the MC1372 supplies a clock signal to the VDG and the rest of the circuit.

The brain behind the whole operation is the 6802 microprocessor chip, which is basically an enhanced 6800. In addition to the features of the 6800, the 6802 includes a built-in 128-byte random access memory, which is used as a scratchpad memory by the Electric Crayon's software. The 6802 is kept busy executing the graphics software, placing picture information in the display memory, feeding control signals to the VDG and directing traffic on the address and data buses.

Memory in the Electric Crayon consists of 1024 bytes of display random access memory (RAM) and 1K of read-only memory (ROM) containing the operating system (dubbed EGOS by Percom), with sockets for adding an additional 5K of display memory, 1K of ROM and 1K of program RAM. All RAM chips are 2114s; two of these are required for each 1K of memory. The 1K of display memory furnished will support the alphanumeric display mode and graphics modes of up to 128×64 picture elements. More memory may be added to support the denser graphics modes (refer to the section on display modes for a discussion of memory requirements).

The graphics software that controls the unit is contained in one 2708 erasable read-only memory (EROM) with a capacity of 1K, and a socket is present for a second 2708 to expand the software. Sockets are also provided for two more 2114 RAM chips, which can be used to contain programs loaded in by the user. These are not, however, at all essential for normal operation.

The display memory is loaded by the 6802 in response to commands entered via the input device. This memory stores control information for each picture element of the display, with each location in memory corresponding to a location on the screen. Depending on the mode selected, each picture element displayed requires from one to eight bits of memory. The VDG scans the display memory and produces a video signal based on its contents. Properly coded display data can also be loaded directly into the display memory from the input device, using the LD* (Load) command of the operating system.



Electric Crayon self-portrait.

Input to the Electric Crayon is through a 6820 peripheral interface adapter (PIA) containing dual I/O ports. One of these ports is monitored by the control program, EGOS, for input commands coming from an attached keyboard or computer. The other half of the 6820 can be hooked up by installing jumpers from points on the printed circuit board to the desired contacts on the rear panel connector. Also, a second 6820 can be plugged in, providing two more ports via a separate connector. However, you must load in your own program to access any port other than the normal input port.

Power is furnished by a regulated supply with dc outputs of 12 V, −5 V and +5 V. A small step-down transformer mounted to the bottom of the cabinet provides isolation from the ac power line.

Display Modes

The Electric Crayon has ten display modes providing alphanumerics, semigraphics and graphics in up to eight colors (green, yellow, blue, red, buff, cyan, magenta and orange) plus black.

Mode 0—alphanumerics/semigraphics. This mode is divided into two subsets, which can appear on the screen simultaneously. The first, alphanumerics, will display any of 64 uppercase letters, numbers and various symbols, punctuation marks and math operators in an 8×12 dot pattern. Colors can be green or orange on a black background, or black on a green or orange background.

The second subset, semigraphics, is a bit more complicated. Each semigraphics display element consists of a rectangular block, with 32 blocks fitting horizontally across the screen and 16 vertically; this gives a total of 512 blocks per screen. In mode 0, each of these blocks is divided into four segments, as shown in Fig. 1a. Within each block, these segments can either be illuminated or extinguished, giving 16 possible patterns in any one of the eight colors (segment selection is described under the commands section of this article). These blocks with appropriate patterns illuminated can be combined to make low-resolution pictures, block letters and so on.

Incidentally, *graph paper* is helpful in constructing a picture with semigraphics blocks, and screen layout pads for the denser graphics modes are available from Percom.

Mode 1—semigraphics. This mode is similar to the four-segment semigraphics of mode 0, except that each

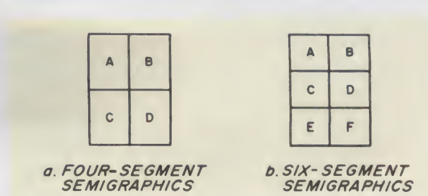
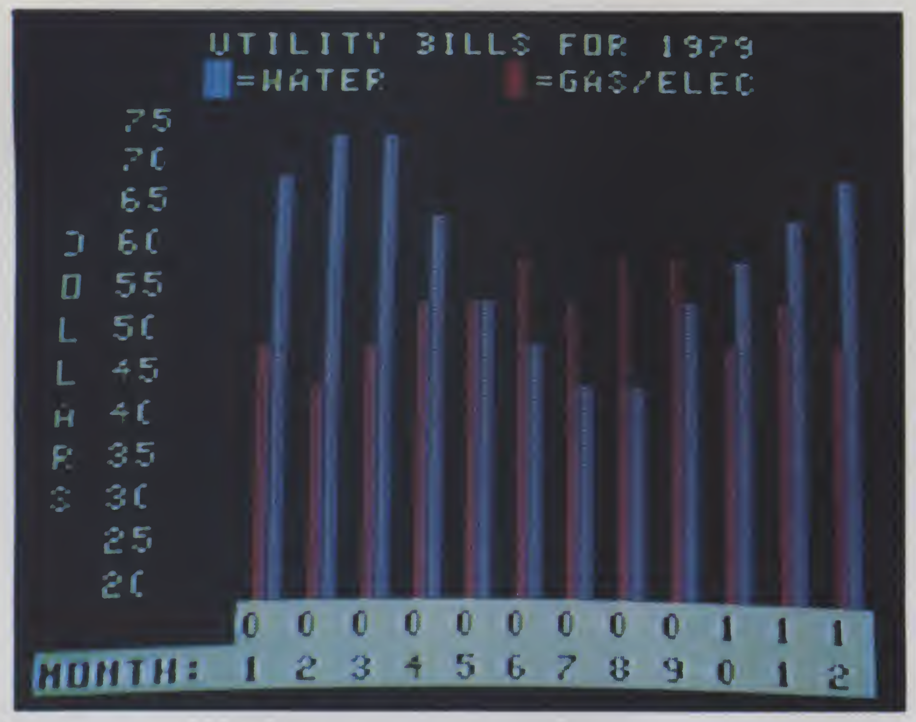


Fig. 1. Semigraphics.



A bar chart in mode 0 semigraphics.

display block is divided into six segments instead of four (see Fig. 1b). Again, there are 32 blocks across the screen and 16 down. The colors can be green, yellow, blue and red on a black background, or, alternately, buff, cyan, magenta and orange on a black background when the Invert command is in effect (refer to the section on commands for details on color selection and inversion procedures).

Mode 2—64×64 graphics. Modes 2 through 9 are full graphics modes, wherein each display element is either on or off, and not subdivided as in the semigraphics modes. Mode 2 provides 64 elements across the screen and 64 down, with each element consisting of four dots horizontally and three screen lines vertically. Colors available in this mode are green, yellow, blue and red on a green background, or buff, cyan, magenta and orange on a buff background in inverse colors. Note that elements displayed in the same color as the background will not be visible, but can be used to erase previously displayed elements of another color to provide animation effects.

Mode 3—128×64 graphics. This mode provides 128 elements across the screen and 64 down. Colors are black and green on a black background surrounded by a green border, or black and buff on black with a buff border. The border is not visible in modes 0 through 2 since it is the same color as the background.

Note that modes 0 through 3 can be implemented using the 1K of display memory provided with the Electric Crayon. Modes 4 through 9 require additional memory as indicated.

Mode 4—128×64 graphics. This display mode has the same resolution as mode 3, but two pairs of four colors are available as in mode 2. You need 2K of display RAM.

Mode 5—128×96 graphics. With this mode, display resolution improves in the vertical direction; colors are identical to mode 3. Again, 2K of RAM is required.

Mode 6—128×96 graphics. Same resolution as mode 5, but with four colors available as in mode 2. This mode requires 3K of display RAM.

Mode 7—128×192 graphics. Better resolution with the same color choices as mode 3. It requires 3K of display memory.

Mode 8—128×192 graphics. This mode provides the same resolution as mode 7, but with the four colors of mode 2. This mode requires 6K of memory.

0—Green	4—Buff
1—Yellow	5—Cyan
2—Blue	6—Magenta
3—Red	7—Orange

Table 1. Color values for the C command.

Mode 9—256×192 graphics. This is the highest resolution mode, with 256 elements across the screen and 192 down. Each element is one dot wide and one scan line high; colors are the same as in mode 3. It also requires 6K of RAM.

The Electric Crayon Graphics Operation System (EGOS)

EGOS is the Electric Crayon's graphics control program, located in the on-board read-only memory. It accepts and processes commands entered via the main input port; these commands consist of ASCII letters and numbers and may be entered directly from a keyboard attached to the port. A more satisfactory approach is to connect the input port to a parallel output port on your computer and send commands to EGOS using PRINT statements in a BASIC-language program. Some hints on doing this will be offered later. Meanwhile, each command will be explained separately.

ERS—Erase command. As the name implies, this command erases the screen of whatever is displayed, leaving only the background color and border, if it is visible.

M—Mode command. This command selects one of the ten display modes and is normally the first command issued when generating a display. The format is Mx, where x is a number 0–9 corresponding to the desired mode. The Mode command should be followed by ERS to clear the screen; reversing this order could leave garbage on the screen. Also, the M command must be followed by another character, such as a space or carriage return, before it takes effect.

C—Color command. The C command selects the colors to be used when generating a display; the resultant colors depend on the mode selected. The format is Cx, where x is a number from 0 to 7 denoting the desired color (see the color values in Table 1).

For mode 0, the color assignments are exactly as shown in the table. For example, entering C2 will produce blue, and C7 orange; in other words, all eight colors are available. If you select a four-color mode (such as mode 1), the only valid color commands are C0, C1, C2 and C3. If the display is in noninverted mode (see the I command), the colors in the left-hand column of Table 1 will be selected with the above commands (i.e., C0 = green, C2 = blue, etc.); if the

```
REM             THIS PROGRAM WILL DISPLAY ALL POSSIBLE PATTERNS FOR
REM             MODE 1 SEMIGRAPHICS, IN FOUR DIFFERENT COLORS.
REM
REM
                LPRINT  "M1 ERS"          REM  SET MODE & CLEAR SCREEN
                Y=0                       REM  INITIAL Y AXIS VALUE
                FOR C=0 TO 3              REM  LOOP THRU COLOR VALUES
                X=0                       REM  INITIAL X AXIS VALUE
                LPRINT "C",C              REM  SET COLOR
                FOR P=0 TO 63             REM  LOOP THRU PATTERN VALUES
                LPRINT "P",P              REM  SET  PATTERN VALUE
                LPRINT "S ",X," ",Y       REM  PLOT THE POINT
                X=X+1                     REM  INCREMENT X AXIS VALUE
                IF X=32 THEN Y=Y+2        REM  IF END OF LINE, GO DOWN
                IF X=32 THEN X=0          REM    2 LINES & RESET X AXIS
                NEXT P                    REM  GET NEXT PATTERN VALUE
                NEXT C                    REM  END OF PATTERNS, DO NEXT COLOR
                END
```

Listing 1. A BASIC program to display all patterns for mode 1 semigraphics.

display is inverted, the colors from the right-hand column will be active (C0 = buff, C1 = cyan, etc.).

Similarly, in modes having only two colors, the only valid color commands are C0 and C1, with C0 producing black (the same as the background) and C1 producing either green or buff for noninverted or inverted displays, respectively. The C command should be issued after the display mode is selected, and will stay in effect until another C command is issued.

Colors can be changed for each element in the display while in the four- or eight-color modes, but all elements in the two-color modes must be the same color. In other words, the entire display must be green on black or buff on black, not a combination of the two. I found the section on color selection in the Electric Crayon manual rather confusing, so I hope this discussion has clarified the subject.

I—Invert command. The I command will change the displayed colors to their complementary colors as explained under the C command; it has no effect in mode 0, which can display all eight colors simultaneously, except to change the color of characters produced with the A or R commands from green to orange. In other modes, this command will invert all colors on the screen, which is to say you cannot use one set of colors on part of the screen and their complements on another. However, interesting blinking effects can be obtained by creating a display and then issuing periodic I commands to reverse the colors.

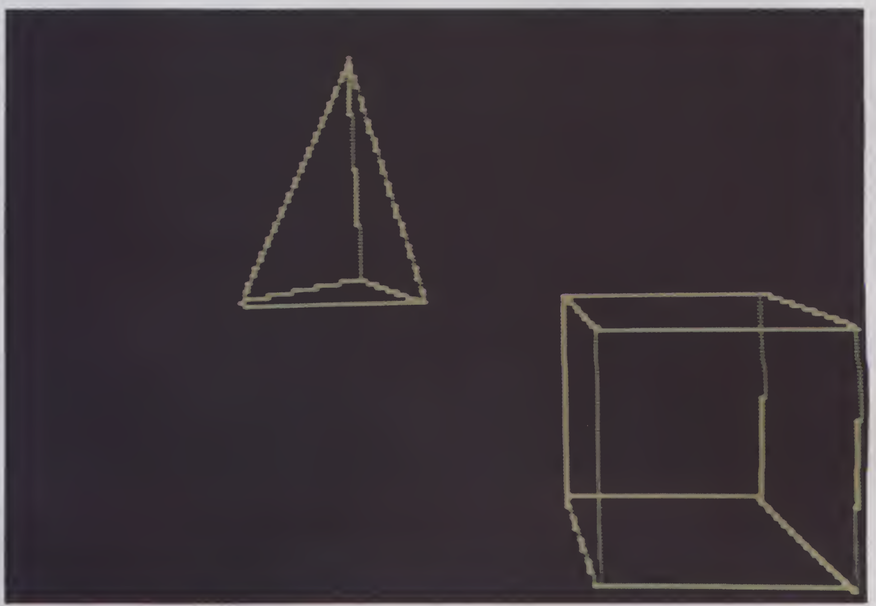
P—Pattern command. This command operates only in the two semigraphics modes and determines which segments of each display block are illuminated. The command is entered as Pn, where n is the code for the particular pattern desired. There are 16 possible combinations for mode 0, with four segments per display block, and 64 for mode 1, with six segments. Space does not permit describing every possible combination, but in both modes 0 and 1, P0 means that all segments will be extinguished, and P15 and P63 mean that all are illuminated for the two modes, respectively.

The easiest way to discover the effects of various values for Pn is to write a simple BASIC program to increment through all possible values and output appropriate commands to the Electric Crayon that will display the results. Listing 1 shows such a program.

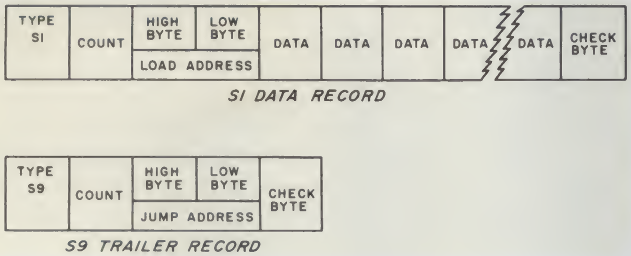
A—Alpha command. The Alpha command displays strings of characters in designated locations on the display screen. This command is valid only for mode 0 and can be used in combination with mode 0 semigraphics. The format is A x y aaa..., where x and y are the desired horizontal and vertical screen coordinates and aaa... is the string of characters to be displayed. For example, the command A 5 8 GRAPHICS will display the word "GRAPHICS" starting at the sixth column of the ninth display line. Valid values of x are 0–31, and 0–15 for y. If the character string is too long to fit on one line, it will be continued on the following line; the command terminates when a carriage return is received. The characters will appear as green on black or orange on black, depending on whether the Invert command is in effect.

R—Reverse Alpha command. This command is identical to the A command, except that the characters appear as black on a green or orange background.

S—Set command. The Set command is used to illuminate a single display element on the screen by specifying the desired x and y coordinates in the format S x y; the valid ranges for x and y vary according to the mode being used, and range from 0 to the maximum number of display elements for that mode in the x or y direction minus 1. When using either of the semigraphics modes, a P command must be issued before the S command to turn on the desired seg-



A 3-D picture in mode 9 using the Sublogic graphics package.



EACH BLOCK REPRESENTS ONE BYTE HEXADECIMAL OR TWO ASCII CHARACTERS

Fig. 2. Motorola S1-S9 record format.

ments of the display element. The resulting value of P will remain in effect for all subsequent S commands until another P command is issued. A C command will determine the color displayed by S commands that follow.

H—Horizontal Line command. Using the format H x y l, the H command will display a horizontal line starting at the specified x and y coordinates for a length of l. The maximum values of x, y and l depend on the mode selected with a prior M command, but any value above the allowable limit will be automatically truncated to the maximum. As with the S command, a P command should be issued first to set the desired element pattern when in the semigraphics modes.

V—Vertical Line command. Same as the H command, except that it displays a vertical line using the command format of V x y l.

LD—Load command.* This command can be used to load machine-language programs into the optional program RAM to provide additional software routines, and can also load properly encoded display data directly into the display memory, causing it to appear on the screen. For example, instead of using the Color and Set commands to illuminate points on the screen, you could load a bit pattern for the desired color into the display memory address corresponding to a specific screen location.

Data entered with the LD* command must be in Motorola S1-S9 ASCII-hexadecimal format, as shown in Fig. 2. All fields in the S1 and S9 records are in ASCII-hexadecimal format, meaning that two ASCII characters are used to represent one byte in hexadecimal notation (i.e., a one-byte field containing F6 in hex would be entered via the input device as F and 6, thereby allowing hexadecimal data to be entered on an ordinary ASCII keyboard). After the LD* command is entered, the Electric Crayon will read S1 and S9 records from the keyboard or computer into memory, and then either begin executing the program that was entered or return to EGOS and wait for more commands.

Data to be entered must be in the S1-S9 record format, wherein each record consists of a type code (S1 or S9), a count field, an address field, a checksum and, in the case of S1 records, a variable-length data field. The first field in a record must be either S1 for a data record or S9 for a trailer record, and the second must be a count field containing the total number of hex bytes in the address field (2), the data (variable) and itself (1). The third field is a two-byte address field, indicating the load address for the following data if in an S1 record, or an address where the Electric Crayon should begin execution if in an S9 record. If an S9 record contains an address field of all zeros, control will return to EGOS. The fourth field in an S1 record contains the data to be loaded; all of the data does not have to be contained in one record, but may be continued in several S1 records. The number of data bytes per record is arbitrary. The final field is the checksum byte, which is used for detecting errors during loading. It contains the complement of the eight-bit sum obtained when all of the data bytes, address bytes and count bytes are added together, with any carries out of the high-order position ignored.

As the records are read into the Electric Crayon, EGOS totals up its own checksum and adds it to the checksum in the input record. If there are no loading errors, the sum should be zero, since the checksum in the record was complemented prior to being output. If an error does occur, EGOS will print "LOAD ERROR" on the display screen and await more commands. Loading stops when an S9 trailer record is read, and control passes to the address in the record or to EGOS if that address is zero.

Hardware Interfacing

If you want to connect the Electric Crayon to a Radio Shack TRS-80 computer, interfacing is a snap; although no connection cable is included with the Electric Crayon, two different cables for the TRS-80 are available from Percom. Both cables let you connect the unit to a TRS-80 in place of a printer, with one cable plugging into a TRS-80 expansion interface and the other connecting to the TRS-80 printer interface cable. In either case, the LPRINT command in BASIC is used to send commands to the Electric Crayon via the printer port. The procedure is thoroughly explained in the manual supplied with the Electric Crayon.

Connecting the unit to a parallel output port on another type of computer is a bit more involved, but still not too difficult. You must first become familiar with the pin assignments for channel A of the Crayon's computer port (Appendix A1 in the Electric Crayon manual). The signals at this port are the eight data lines, D0 through D7, which accept data from the external computer, and two control lines labeled $\overline{\text{STROBE}}$ and BUSY.

The first, $\overline{\text{STROBE}}$, is an input signal sent from the external device's output port indicating that valid data is present on the data lines. Most popular I/O boards for various computers have this signal present on each parallel output port; this signal must be negative true; that is, it goes low to indicate data present.

The second line, BUSY, is an output signal from the Electric Crayon to indicate that it cannot accept input data and that the computer should wait until the line goes low before sending anything. This signal should be connected to one data line of an input port that is used as a status port by the computer; again, most I/O boards include a status port. If your computer's parallel output port has an acknowledge input for accepting a handshaking signal from the external device, it should be enabled by tying it to +5 V dc. A simple program that waits to transmit data to the Crayon until the BUSY line is low is shown in Listing 2, written in 8080 assembly language.

Mode 9 graphics—a Lissajous figure plotted with a BASIC program.

Although the manual gives pin numbers on the rear panel connector for each signal, it makes no mention of how the pins are located on the connector. Viewed from the rear of the chassis, the computer port connector is on the right, with the auxiliary port connector on the left. The computer port connector is a card-edge type having 17 pins on each side, with pin 1 being the leftmost pin on the underside of the board. Pin 2 is above pin 1 on the top of the board; thus, all odd-numbered pins are on the bottom and even-numbered pins on top.

A suitable connector is the AP Products 924064-36-R, which comes attached to 18 inches of ribbon cable and is available from most large mail-order parts outlets. Any similar connector with 0.100-inch pin spacing should also work. The auxiliary port requires a 40-pin card-edge connector, also with 0.100-inch spacing. Pins 31, 32 and 33 on the computer port connector are grounded, and one of these should be connected to a ground on the computer.

Alternately, a bare-bones graphics system can be set up by attaching an ASCII keyboard to the Electric Crayon computer port. Simply attach each of the data lines from the keyboard to the corresponding line on the port and connect the keyboard $\overline{\text{STROBE}}$ output to the $\overline{\text{STROBE}}$ input on the port. You can then type commands to the Electric Crayon on the keyboard. While this method is not as elegant as attaching a computer, it is a good start.

Video Connections

If you have a video monitor, connection to the Electric Crayon requires only an F59 connector, a length of RG-59/U coax cable and a connector to fit the video input on the monitor. Instructions are furnished for converting the Electric Crayon to output a UHF signal so that it can be connected to the antenna terminals of a TV set, but this requires cutting components from the circuit board and would be difficult to reverse.

Instead, I recommend using an inexpensive rf modulator such as the Pixie-verter by ATV Research, which connects between the video output on the Electric Crayon and the antenna terminals on the TV set. This lets you use either a TV set or a monitor, the latter producing higher resolution with the denser graphics modes. A monitor also will eliminate interference caused by stray signals from the computer that tend to sneak in through the antenna terminals and garble the picture.

If you do use a modulator, be sure to disconnect the antenna from the TV set to prevent broadcasting your signal to other nearby sets; you might think computer graphics are fascinating, but your neighbors may not share your enthusiasm!

Software Interfacing

If you have a TRS-80 and have connected the Electric Crayon to its printer port according to the instructions in the manual, all that you have to do to talk to the Crayon is use the LPRINT BASIC statement to send commands to the printer port. For example, the statement LPRINT "ERS" would erase the display screen.

If your computer is not a TRS-80, things get a little stickier. The easiest approach is to use a version of BASIC that includes an LPRINT statement and then connect the Electric Crayon to the printer port. This approach is a bit inconvenient, however, if you have a printer and must switch devices whenever something needs to be printed.

A more elegant approach is possible if your BASIC includes a POKE statement and you know where the I/O driver routines are located in memory. Connect the Crayon to a separate port, then include a BASIC subroutine in your program to POKE the Electric Crayon port numbers and status port mask bits into the I/O driver routine for the printer, so that output from the LPRINT statement goes to the Crayon port instead of the printer. A similar subroutine would POKE the original values back in to access the printer (see Listing 3 for an example of this method).

```
                    *       THIS ROUTINE IS CALLED WITH THE CHARACTER TO BE SENT
                    *       TO THE ELECTRIC CRAYON IN REGISTER B.
                    *
                    *
1000 DB05           SEND:   IN      05      ;READ CRAYON STATUS PORT
1002 E680                   ANI     80H     ;IS THE CRAYON BUSY?
1004 C20010                 JNZ     SEND    ;YES, WAIT
1007 78                     MOV     A,B     ;GET CHAR. TO BE SENT
1008 D308                   OUT     08      ;SEND IT TO CRAYON
100A C9                     RET             ;RETURN TO CALLER
```

Listing 2. Routine that waits for the Crayon's busy line to go low before sending a character.

Desk Main/Frame

```
                *       THIS ROUTINE IS CALLED WITH THE CHARACTER
                *       TO BE PRINTED IN REGISTER C
                *
                *
7E36 DB03       LIST:   IN      03      ;READ STATUS PORT
7E38 E601               ANI     01      ;CHECK STAT FOR READY TO PRINT
7E3A CA367E             JZ      LIST    ;NOT READY, WAIT
7E3D 79                 MOV     A,C     ;GET CHARACTER
7E3E D302               OUT     02      ;OUTPUT IT TO PRINTER PORT
7E40 C9                 RET             ;RETURN TO CALLER
```

A. The Standard CP/M Line Printer Routine

```
REM     THIS ROUTINE MODIFIES THE CP/M LINE PRINTER
REM     OUTPUT ROUTINE TO ROUTE OUTPUT FROM LPRINT
REM     STATEMENTS IN CBASIC TO THE ELECTRIC CRAYON
REM
        POKE    7E37H,05H       REM  CRAYON STATUS PORT
        POKE    7E39H,80H       REM  CRAYON STATUS BIT MASK
        POKE    7E3AH,0C2H      REM  INVERT JMP INSTRUCTION
        POKE    7E3FH,08H       REM  CRAYON OUTPUT PORT
```

B. The BASIC Statements Needed To Modify The Routine

```
                *
                *       THIS ROUTINE IS CALLED WITH THE CHARACTER
                *       TO BE SENT TO THE CRAYON IN REGISTER C
                *
7E36 DB05       LIST:   IN      05      ;READ ELECTRIC CRAYON STATUS PORT
7E38 E680               ANI     80H     ;IS THE CRAYON BUSY?
7E3A C2367E             JNZ     LIST    ;YES, WAIT
7E3D 79                 MOV     A,C     ;GET CHARACTER
7E3E D308               OUT     08      ;OUTPUT IT TO CRAYON
7E40 C9                 RET             ;RETURN TO CALLER
```

C. Source Listing Of The Result

Listing 3. Using the POKE statement in BASIC to modify the line printer driver.

If your BASIC has POKE but no LPRINT statement, use the above scheme to modify the console output routine so that output from the PRINT statement will be sent to the Electric Crayon. A similar approach can be applied to other languages, such as FORTRAN. Of course, commands can be sent to the Crayon from assembly-language programs, but numeric values (such as x-y coordinates) must be converted to ASCII character strings for transmission to the unit. For versatility and easy implementation, BASIC is the way to go.

Observations

Having used the Electric Crayon for several weeks with a variety of applications, my experiences thus far have been quite favorable. I phoned my order directly to Percom (they accept credit cards), and received the unit in about five weeks.

After reading the manual and connecting the Crayon to my monitor, I threw the power switch and rejoiced as the sign-on message flashed on the screen. I then spent several hours connecting the unit to a parallel port on my computer, a souped-up Altair 8800, and writing some simple software driver routines.

As I mentioned, the manual is a bit vague concerning interfacing to anything other than a TRS-80, so some trial and error was necessary. The manual does contain a good section on familiarization with the EGOS commands and includes some simple demonstration programs. I have had good results driving the Electric Crayon with programs written in CBASIC using the CP/M operating system, by poking modifications into the CP/M printer output routine as shown in Listing 3. The Crayon's semigraphics modes are ideal for drawing colorful bar charts and graphs using values calculated with BASIC programs, while mode 9 is great for plotting equations. I have also had excellent results driving the Electric Crayon with the BASIC language 3D graphics package from Sublogic. The package is well-documented and, coupled with the Crayon, is a very good tool for learning computer graphics concepts.

I found fault with only two aspects of the Electric Crayon's performance. The first is that the ERS command flashes garbage on the screen for an instant before erasing the display. This is only a minor annoyance unless you want to erase the screen between frames of an animated display. The second is that the time required to draw lines using the H and V commands in the denser graphics modes is rather high; several seconds are required to draw long lines in mode 9, which is bad news for animated displays.

One solution is to load the display data directly into the Electric Crayon's display memory with the LD* command, producing an almost instantaneous result. The catch is that you have to write some rather complex assembly language routines to calculate load addresses, data content and checksum, and convert everything to ASCII-hexadecimal format for transmission to the Electric Crayon.

Conclusion

To sum up, I am generally pleased with the Electric Crayon; it's a good buy for the money. The documentation is well-written and, with the aforementioned exceptions, complete and helpful. The schematic diagram and theory of operation provided help unravel the mysteries of the Crayon's innards, and the source listing of EGOS is invaluable for the software hackers. The unit is powerful enough to be a stand-alone graphics computer with its four I/O ports and provision for adding program and read-only memory. With a little imagination, the possibilities are almost endless. Happy coloring! ■

Additional Resources

- A 3D Microcomputer Graphics software package written in BASIC as well as various assembler languages is available from Sublogic, Box V, Savoy, IL 61874. The cost is quite low and the extensive documentation does a good job of explaining the basics of computer graphics.
- The Pixie-verter from ATV Research, 13-B Broadway, Dakota City, NE 68731, is a good rf modulator for use with the Electric Crayon. It costs about $10 and is available from many dealers.
- *The TV Typewriter Cookbook* by Don Lancaster, published by Howard Sams, contains detailed instructions on adding a direct video *input* to an ordinary TV set.

Using the LD* Command

This example explains the S1 and S9 records required to draw a line on the display screen by loading picture data directly into the Electric Crayon's display memory. Specifically, a blue line eight blocks long will be drawn across the top of the screen using mode 1 semigraphics. Under mode 1, one byte of display memory is needed for each display block on the screen, since both color and semigraphics pattern information must be present. This data is encoded in each mode 1 display byte as follows:

$C_1\ C_0\ P_5\ P_4\ P_3\ P_2\ P_1\ P_0$

where C_1 and C_0 select the color, and P_5–P_0 select which of the six segments of each semigraphics block are illuminated. This information is contained in the Motorola applications notes for the MC6847, available from Motorola distributors.

In this example, we will select blue with all segments illuminated, which is encoded like this:

1 0 1 1 1 1 1 1

Therefore, each of our bytes in display memory will contain 10111111 in binary, or BF in hexadecimal.

We want to draw our line across the top of the screen starting at the leftmost position for a length of eight display blocks, which will require eight bytes in the Crayon's display memory. According to the Electric Crayon manual, the display memory starts at address D000 in hex, beginning in the upper left-hand corner of the display. Since there are 512 block positions on the screen in mode 1, and each block requires one byte of memory, the last position on the screen is at address D1FF. This block is in the lower right-hand corner of the screen. Our eight-block line will use addresses D000 through D007.

We can now begin building our S1 data record. The first two bytes are the record type, S1. The second two are the count field, containing the number of *hex* bytes encoded in the count, address and data fields. In our example there are eight data bytes, plus two bytes for the address field and one for the count field itself. Thus, we have 8 + 2 + 1 = 11, or 0B in hex. After converting to ASCII, the characters 0 and B are placed in this field. The next field in the record is the load address, which in this case is D000; the four ASCII characters D000 are inserted here. The eight data bytes are next, again expressed in ASCII as B and F repeated eight times, for a length of 16 bytes.

Now for the hard part: the checksum. It is calculated by adding together the data bytes, address bytes and count byte as an eight-bit sum, with carries out of the high-order bit ignored; the total is then complemented by making each zero bit 1 and each one bit 0, converted to two ASCII characters and placed in the record. Note that the hexadecimal values for the fields are added together before they are converted to ASCII characters.

In our example, we will add the count byte (0B), the address bytes (D000) and the eight data bytes as follows:

0B + D0 + 00 + BF + BF + BF + BF + BF + BF + BF + BF

giving a total of D3 if high-order carries are dropped. The complement of D3 is calculated to be 2C by inverting each bit. This value is converted to ASCII 2 and C and placed in the record. The checksum is most easily calculated using an assembler language routine that zeros an eight-bit accumulator, adds each byte to it, and then takes the complement (the 8080 assembler language instruction to do this is CMA, with the Z-80 equivalent being CPL). The complete ASCII S1 record appears as follows, with a length of 26 bytes:

S10BD000BFBFBFBFBFBFBFBF2C

The same rationale is used to construct the count, address and checksum for the S9 trailer record; the zero address field returns control to EGOS after the record is read. The S9 record appears below:

S9030000FC

The commands that need to be entered to draw our line are as follows:

```
M1      (Set Mode 1)
ERS     (Clear Screen)
LD*     (Load)
S10BD000BFBFBFBFBFBFBFBF2C
S9030000FC
```

Although this method is much more cumbersome than using EGOS commands to draw displays, it is faster and is ideally suited to assembler language driver programs.

Base 2's dot matrix printer gives you the flexibility to create and use your own custom character sets.

Dots Incredible!

By James J. Conroy

"You, sir!—define a character!" For the computerist who is lucky enough to have a dot-matrix printer, this command may just be the key to hard copy that ranges from graphics to foreign language scripts.

The dot-matrix printer system arrived on the scene some years ago and brought a new dimension to hard copy: user-defined character (UDC) sets.

Unlike a fixed-type element printer (such as a typewriter with type bars or a type ball), the dot-matrix mechanism uses a single moving print head with an array of holes, through which solenoid driven wires can protrude. By intelligent selection of the impact of these wires, the entire American Standard Code for Information Interchange (ASCII) character set is produced. Typically, the matrix of the print head has a dimension of 5 by 7 or 7 by 9. This means that the print head has one bank of seven wires that prints five columns to form a character (or for the 7 by 9 matrix, nine wires and seven columns). The permutations—or possible combinations—of dot configurations that could conceivably be produced are described in Table 1.

That's a lot of possible characters, and, for the most part, we are only using a subset of 96 characters—the ASCII set! So what about all the rest of the character configurations? Well, some printer manufacturers have opened the door to using the whole matrix by letting the programmer access and control the print head.

How It Is Done

The manner in which a programmer is empowered to produce user-defined characters can vary from one machine to the next. For our purposes, we will consider the methodology set forth for the new Base 2 printer. This compact and versatile unit is one of the most pleasing combinations of low cost and high performance I have seen. In addition to the options available to enhance the performance of this little marvel, the UDC feature comes as a standard item. (As do the full upper/lower ASCII set, elongated characters, immediate interface capability for the four most popular interface modes and a host of other desirable features.)

Fig. 1. A wire can impact and produce a dot in any or all of the positions.

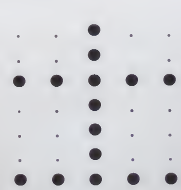


Fig. 2. This design produces a "plus or minus."

$5 \times 7 = 35$ dots (on or off) = 2 to 35th power = 3.43597×10^{10}
$7 \times 9 = 63$ dots (on or off) = 2 to 63rd power = 9.22339×10^{18}

Table 1.

As with most other dot-matrix printers, the Base 2 uses resident software (usually referred to as firmware) to define and control the print head's configuration of the print characters. In fact, provisions are included in its design for the easy installation (via IC sockets) of additional EPROM chips to boost the number of on-board character sets to anywhere from four to eight sets. Base 2, Inc. (1835A Dawns Way, Fullerton, CA 92631), currently offers four sets in EPROM and plans for more.

To create and use your own character set, the following steps must be taken:

- Design and define the characters you want to produce. The result is a series of data statements which must be transmitted to the printer.
- Transmit the data to an allocated memory in the printer (RAM).
- Switch to your user-defined characters (when required) by a special control code.
- Switch back to standard characters (by control code) as necessary.

Let's go into a little more detail for each step to see how easy it is.

Design and Define

It was a pleasant surprise to find out how simple it is to design a character for the Base 2 printer. Begin by envisioning the print matrix configuration. Basically, it looks like Fig. 1. Each dot represents a position where a wire can strike to produce an impression. Therefore, you must decide which pattern of impressions will create an image of your desired char-

Address correspondence to James J. Conroy, 57 E. Garrison St., Bethlehem, PA 18018.

```
35 REM THIS SETS PRINTER UP FOR FEEDING DATA
40 LPRINT CHR$(27);CHR$(75)
45 REM THIS READS AND FEEDS DATA
50 FOR C=1 TO 48
60 FOR A=1 TO 10
70 READ X
75 X=X+128
80 LPRINT CHR$(X);
90 NEXT A
100 RESTORE
110 NEXT C
115 REM THIS TESTS STANDARD CHARACTERS
120 LPRINT"ABCDDEFGHIJ"
130 LPRINTT CHR$(27);CHR$(76)
135 REM THIS TESTS UDC
160 LPRINT "ABCDEFGHIJ"
500 DATA 0,0,0,0,0,68,68,127,68,68
```

Listing 1.

```
01000100
01000100
01111111
01000100
01000100
```

Fig. 3. Ones now represent dots, and zeros represent spaces.

```
1000100
1000100
1111111
1000100
1000100
```

Fig. 4. An extra zero has been added at the bottom of each column to make eight-bit binary numbers.

acter.

For example, consider the design of the non-ASCII character for the mathematical term "plus or minus." In Fig. 2, the appropriate dots have been marked to produce this symbol. To encode this into a data statement that will define this symbol for the printer to store in its memory, you can imagine the same design with zeros for blank spaces and ones for dots. (See Fig. 3.)

Does this remind you of anything? What we have now are five columns that represent binary numbers! This is the key to forming the necessary data statement to define this character. It will be convenient to consider each column as an expression of an eight-bit binary number. Bit number 8 will always be zero (imagine one more zero under each column). If you turn Fig. 4 90 degrees clockwise, you can easily see the binary numbers which define this character. Since the Base 2 printer has built-in provisions for one-dot intercharacter spacing, you need only concern yourself with the construction of the character itself.

To actually write the data statement, you must consider the decimal equivalent of these five binary numbers. It turns out that for this design, you really only have two different numbers. They are

01000100 = 68
01111111 = 127

Therefore, the data statement should read:

```
500 DATA 68,68,127,68,68
```

This will define your single character.

Transmit

You are now ready to transfer the information in the data statement to the printer. The printer will store the data in its RAM, which is set aside for this purpose. To transmit the character definitions (or, for that matter, *any* information or command) to the Base 2, employ the LPRINT CHR$(X) command from BASIC. In my talks with the Base 2 people (who were quite helpful and courteous), I learned that the BASIC software control codes may vary from one computer to the next. What I am relating are the commands for a Base 2/TRS-80 interfacing. The general idea is the same for other Base 2/computer interfaces.

You must always advise the Base 2 printer that a command or information is being sent. The command code for this is LPRINT CHR$(27). This must *always precede* any instruction which you are going to send. The next command—LPRINT CHR$(75)—is the code that tells the Base 2 that you are about to load data into the UDC/RAM. These two commands would be expressed in the BASIC statement as:

```
40 LPRINT CHR$(27);CHR$(75)
```

Now you want to transmit the character defining data, so you must READ the data statement and LPRINT the data values in CHR$(X) form. A sample program might be:

```
50 For A=1 TO 5
60 READ X
70 LPRINT CHR$(X)
80 NEXT A
500 DATA 68,68,127,68,68
```

You are almost there. You must take into consideration one more factor. When implementing the UDC feature of the Base 2 printer, you cannot just send five elements of data down the line to the printer memory. You must fill up the entire UDC/RAM, even if you don't have that many characters to define. There is space in the memory for 96 characters (at five data elements per character), so you must send 480 elements of data to the printer. The logical choice is to fill the remaining area in the memory with data that defines the "space" character. In our situation, it might be nice to have spaces in between our "plus or minus" character (for test purposes), so you can change the data statement to read:

```
500 DATA 68,68,127,68,68,0,0,0,0,0
```

This defines your designated character and a "space." You should further modify the program to transmit this data (for two characters) 48 times. Remember: 48 × 2 characters (five data elements each) = 96 characters (480 elements of data). The program now reads:

```
40 LPRINT CHR$(27);CHR$(75)
50 REM WE HAVE TO FILL UDC/RAM
   WITH 96 CHARACTERS
60 FOR C = 1 TO 48
70 REM WE NOW HAVE 10 DATA ELE-
   MENTS
80 FOR A = 1 TO 10
90 READ X
100 LPRINT CHR$(X)
105 NEXT A
110 RESTORE
115 NEXT C
500 DATA 68,68,127,68,68,0,0,0,0,0
```

Use It!

You have designed, defined and

```
ABCDEFGHIJ
±  ±  ±  ±  ±
```

Fig. 5. The final printout.

stored your very own character. Now use it! To do this, you send the command to the printer which enables the UDC set. (Remember, you have a whole set of these buggers!) It is:

LPRINT CHR$(27);CHR$(76)

Previous to this command, you are drawing standard characters from the printer's ROM. Now you will be drawing from the UDC/RAM. A good way to see the difference would be to add the following statements to your program:

```
120  LPRINT "ABCDEFGHIJ"
130  REM ENABLE UDC SET
140  LPRINT CHR$(27);CHR$(76)
150  REM THIS SHOULD GIVE US OUR
       "PLUS OR MINUS"
160  LPRINT "ABCDEFGHIJ"
```

Fig. 5 is the printout from the Base 2, displaying our pride and joy. When defining a whole set of different characters, you might, at this point, have to go back and rework your designs. A little trial and error, along with visual comparisons, will finally yield a pleasing end product.

Listing 1 is the program which accomplishes our goal.

Switch

Now you have the character(s) and you can call the UDC set when you need it. Don't forget to switch back to the standard ASCII set for your other output. This is accomplished by the command:

LPRINT CHR$(27);CHR$(77)

If you remember your switching commands, you can merrily bounce back and forth from standard to UDC to your heart's content. Yes, Virginia, there is even a command to wipe out your creation from the RAM:

LPRINT CHR$(27);CHR$(74)

Conclusion

By now, visions of possibilities must be popping into your head. What can you define? Almost anything. I'm working on an astrological character set. Next, I'll probably create a UDC set that matches the graphics blocks of the TRS-80 video display. If you come up with something worthwhile, you might want to send a copy to Base 2, Inc. They can even burn an EPROM that will give you your set in a permanent mode. ■

Puts you in control of your data files.

Database Manager For the North Star

By John E. Bailey

I've seen a number of database systems described or advertised in different magazines. In each case, I wondered if the system would do what I wanted it to do, or if I would have to pay a premium price for the few simple functions I was looking for.

To avoid such an unpleasant situation, I decided to write my own.

```
10 REM  *****  DATABASE DEFINITION  *****
20 REM
30 REM  CODED BY JOHN E. BAILEY
40 REM  SULPHUR, LA   APRIL, 1980
50 REM
60 REM  *****  PROGRAM NAME IS  "DBDEFN"  *****
70 REM
80 P1=11
90 FOR I=1 TO P1\!\NEXT
100 DIM L$(15)
110 !"      *****  DATABASE DEFINITION  *****"
120 !
130 INPUT "      ENTER NAME OF DATABASE (4 CHARACTERS) ? ",N$
140 IF LEN(N$)<>4 THEN 120
150 F$=N$+"DEFN,2"
160 INPUT "      DO YOU WANT ME TO CREATE THE DEFN FILE (Y OR N) ? ",Z$
170 IF Z$<>"Y" THEN 190
180 CREATE F$,4
190 OPEN #1,F$
200 L=0
210 IF Z$="Y" THEN 230
220 READ #1 %0,L
230 INPUT "      ENTER TOTAL RECORD LENGTH ? ",R
240 INPUT "      ENTER START POSITION OF KEY ? ",K0
250 INPUT "      ENTER STOP POSITION OF KEY ? ",K9
260 INPUT "      ENTER POSITION OF DELETE FLAG ? ",D9
270 WRITE #1 %0,L,N$,R,K0,K9,D9
280 FOR I=1 TO 11\!\NEXT
290 !"      BEGIN RECORD DEFINITION PHASE ....."
300 !
310 !"      YOU MAY ENTER UP TO 15 CHARACTERS FOR EACH LABEL."
320 !"      YOU MUST ALSO ENTER THE START AND STOP POSITION"
330 !"      FOR EACH FIELD IN THE RECORD."
340 !
350 I=1
360 L$="               "
370 !"      ENTER LABEL",I," ? ",
380 INPUT "",L$(1,15)
390 IF L$(1,3)="END" THEN 480
400 !"      ENTER START POSITION FOR ",L$," ? ",
410 INPUT "",S
420 !"      ENTER STOP POSITION FOR ",L$," ? ",
430 INPUT "",P
440 WRITE #1,L$,S,P
450 I=I+1
460 !
470 GOTO 360
480 CLOSE #1
490 FOR I=1 TO P1\!\NEXT
500 INPUT "      DO YOU WANT ME TO CREATE THE DATA FILE (Y OR N) ? ",Z$
510 IF Z$<>"Y" THEN 560
520 !\!\INPUT "      HOW MANY LINES ITEMS IN DATABASE ? ",P
530 D$=N$+"DATA,2"
540 M=INT(R*P/256)+4
550 CREATE D$,M
560 !\!\!"      *****  END OF DATABASE DEFINITION PHASE  *****"
570 !\!
580 END
```

Listing 1. Database definition program DBDEFN.

I took several factors into consideration when designing my system. I didn't want a database that had to have records with unique keys. I also didn't want to be concerned with keeping the records on the database in any particular sequence. To do this, I had to sacrifice access time, since each database request involves a sequential search. But most of the databases I work with are relatively small, so I can live with this longer access time.

I also wanted a system where I could access any number of different databases with one program. I wanted to be able to add records, delete records, update records, search for records by key, scan for records containing a given character string and retrieve records by record number.

All of these considerations led to the development of the database manager system for the North Star described here.

Defining the Database

First, I'll define a hypothetical database to use as an example. Suppose I want a catalog of all the spare parts in my parts box: resistors, capacitors, transistors, integrated circuits and so on. I want each record in the catalog to contain the part number, a description of the part and how many I have on hand.

The part number will be ten characters long. The first two characters of the part number will be type indicator—IC for integrated circuit, TR for transistor, RS for resistor, CP for capacitor and so forth. The other one to eight characters will be the *actual* part identifier. Thus, the part num-

John E. Bailey, 1100 Post Oak 1, Sulphur, LA 70663.

bers will look like this: IC7404, RS220K and CP.02MF.

I'll also let the part number be the record key. Each record will contain a 15-character description and two characters to indicate the quantity on hand from 00 to 99. I also need to define a one-character field to be the delete flag. The part number will be in position 1 through 10 of the record, the description in position 11 through 25, the quantity on hand in position 26 and 27 and the delete flag in position 28. The total record length will be 28 characters.

The database consists of two files, one containing the database definition and the other the data. Each database will have a four-character name. For my parts database, the name will be PRTS. The database definition file will be named PRTSDEFN, and the data file, PRTSDATA. Later you'll see how these names are created by the database programs.

The definition file consists of two sections—the header and the body. The header contains the record number of the next available record, the database name, the record length, the key location and the delete flag location. The definition body contains a variable number of entries. Each entry contains a 15-character label, a start position S and a stop position P. The label names the field that starts in the record at location S and ends at location P. Fig. 1 shows the relationship between the database definition and the database record for the PRTS database.

DBDEFN is the program that creates and maintains the database definition file (Listing 1). P1 is a constant used in each of the programs in the system to define the number of lines on your CRT terminal less 1. The program will ask for a four-character database name, in my case PRTS, which is prefixed to DEFN to form the name of the database definition file.

If you are running a single disk system, remove the ,2 in line 150. If you are defining a new database, respond Y to line 160. If you want to modify an existing database definition, reply N. Answer each of the questions presented. Type END for the label to end the definition phase. If you want the program to create the data file, respond Y to line 500. The program will create a data file big enough to hold the number of line items you supply in line 520. The name of the data file is the name given in line 130 suffixed by DATA. See Sample 1 for an example of the database definition.

```
*****  DATABASE DEFINITION  *****

ENTER NAME OF DATABASE (4 CHARACTERS) ? PRTS

DO YOU WANT ME TO CREATE THE DEFN FILE (Y OR N) ? Y


ENTER TOTAL RECORD LENGTH ? 28
ENTER START POSITION OF KEY ? 1
ENTER STOP POSITION OF KEY ? 10
ENTER POSITION OF DELETE FLAG ? 28



BEGIN RECORD DEFINITION PHASE.....

YOU MAY ENTER UP TO 15 CHARACTERS FOR EACH LABEL.
YOU MUST ALSO ENTER THE START AND STOP POSITION
FOR EACH FIELD IN THE RECORD.


ENTER LABEL 1 ? PART NUMBER
ENTER START POSITION FOR PART NUMBER ? 1
ENTER STOP POSITION FOR PART NUMBER ? 10

ENTER LABEL 2 ? DESCRIPTION
ENTER START POSITION FOR DESCRIPTION ? 11
ENTER STOP POSITION FOR DESCRIPTION ? 25

ENTER LABEL 3 ? QTY ON HAND
ENTER START POSITION FOR QTY ON HAND ? 26
ENTER STOP POSITION FOR QTY ON HAND ? 27

ENTER LABEL 4 ? END


DO YOU WANT ME TO CREATE THE DATA FILE (Y OR N) ? Y


HOW MANY LINES IN THE DATABASE ? 150



*****  END OF DATABASE DEFINITION PHASE  *****
```

Sample run 1. How a database is defined using DBDEFN. The database being defined here is the example PRTS database.

After defining the database, it must be formatted. This is done by DBFRMT (Listing 2). Entering the name of the database and this program will propagate blanks throughout the entire data file. The actual number of records in the database will be slightly greater than requested in DBDEFN. The next record pointer in the database definition is set to zero. Therefore, this program can be used to reformat an existing database. Normally, though, it would only be run after creating a new database.

Using the Database

After these two steps, the database is ready to use.

First, of course, I load records into the database, using the database

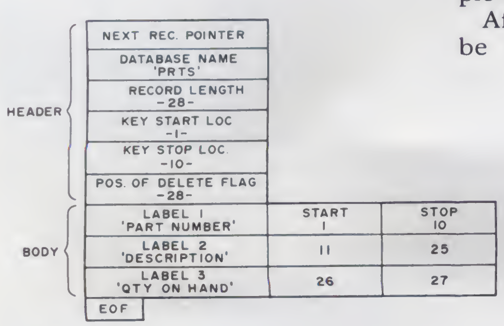


Fig. 1a. Organization of database definition file.

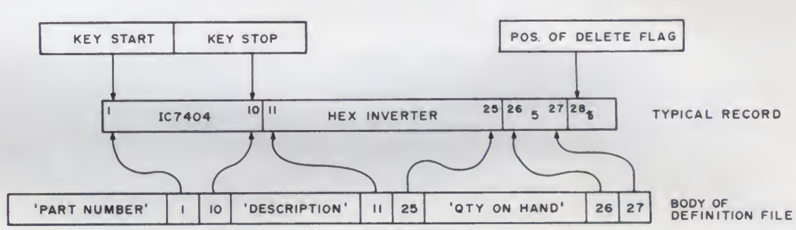


Fig. 1b. Relationship between a record and the database definition file.

manager program DBPROG (Listing 3). It supervises all of the database activities, and is controlled by a menu screen.

Load DBPROG and run it. Type in the database name, and the master menu will appear. Select option 1 to add records to the database. The program asks for the information by label as the record is defined in the database definition.

At this point you can add as many records as you want. Remember my previous comments on keys for the example database. If you want to group certain items together, all or the first few characters of the key should be common. For example, all integrated circuits may have a key beginning with IC, resistors beginning with RS and so on. Each time you add a record, the next record pointer in the database definition is incremented by one. Type END at any point to end this phase and return to the menu. Sample 2 shows how records are added to the database.

Records can be added at any time. The location of the next available database record is maintained in the database definition header. If you return via option 1 to add records, the first one will be added at the record that the next record pointer indicates.

Select option 2 to search the database by key. Enter the key you want to search for. You need not enter the entire key. For example, if you enter IC, all records whose keys begin with IC will be retrieved (see Sample run 3). Thus, you can retrieve groups of records with only one access to the database. You can select an alternate key by replying Y to line 740. Enter the starting position in the record of the alternate key. The default key location is reset after each complete access. You may display the entire database by typing ALL for the key.

Records may be routed to the CRT or to the printer. If routed to the CRT, the program pauses after each record is displayed. Press return to continue or type END to stop. If the output is to the printer, there is no pause and the print continues until the end of the database is reached.

You can scan the database for a character string by selecting option 3. Enter the character string you want to scan for. Each record that contains that string will be displayed. The scan process is much slower than the key search and should be used only if a record cannot be retrieved by its key. As before, the program pauses after each record is displayed, and you can type END to stop.

The fastest way to retrieve a record is by record number. You can do this only if you know the physical number of the record you want. Select option 4, enter the record number and it will be retrieved immediately. Record numbers start at 0 and must not be greater than the next record pointer less 1.

```
10 REM  *****  FORMAT DATABASE  *****
20 REM
30 REM  CODED BY JOHN E. BAILEY
40 REM  SULPHUR, LA   APRIL, 1980
50 REM
60 REM  *****  PROGRAM NAME IS  "DBFRMT"  *****
70 REM
80 P1=11
90 FOR I=1 TO P1\!\NEXT
100 !"      *****  FORMAT DATABASE  *****"
110 !
120 INPUT "      ENTER NAME OF DATABASE ? ",N$
130 IF LEN(N$)<>4 THEN 110
140 F$=N$+"DEFN,2"
150 OPEN #1,F$
160 READ #1 %0,L,N$,R
170 L=0
180 WRITE #1 %0,L,NOENDMARK
190 CLOSE #1
200 DIM R$(R)
210 R$=""
220 FOR I=1 TO R\R$=R$+" "\NEXT
230 D$=N$+"DATA,2"
240 OPEN #1,D$,L
250 L=INT(256*L/(R+2))-2
260 !\!"      THERE ARE",L," RECORDS IN THE DATABASE."
270 !
280 FOR I=1 TO L
290 WRITE #1,R$
300 NEXT
310 CLOSE #1
320 !"      *****  FORMATTING OF DATABASE IS COMPLETE  *****"
330 !\!
340 END
```

Listing 2. Database formatting program DBFRMT.

```
                    *****  DATABASE MANAGER  *****

                  SELECT ONE OF THE FOLLOWING OPTIONS:

             1 - ADD RECORDS            5 - UPDATE A RECORD
             2 - KEY SEARCH             6 - DELETE A RECORD
             3 - CHARACTER SCAN         7 - DISPLAY DBDEFN
             4 - RECORD NUMBER          8 - SWITCH DATABASES

                               ENTER YOUR SELECTION ? 1


     PARTS NUMBER    ? IC7400
     DESCRIPTION     ? QUAD NAND GATE
     QTY ON HAND     ? 4

     PARTS NUMBER    ? IC7402
     DESCRIPTION     ? QUAD NOR GATE
     QTY ON HAND     ? 5

     PARTS NUMBER    ? RS220K
     DESCRIPTION     ? 220K OHM RES
     QTY ON HAND     ? 10

     PARTS NUMBER    ? CP.01MF
     DESCRIPTION     ? .01 MF CAP
     QTY ON HAND     ? 2

     PARTS NUMBER    ? IC7408
     DESCRIPTION     ? QUAD AND GATE
     QTY ON HAND     ? 7

     PARTS NUMBER    ? END
```

Sample run 2. How to add records to the database (option 1). The program will prompt you to supply information as the record is defined in the database definition. Add as many records as wanted. Type END to return to the master menu.

✓Reader Service—see page 178

You can update or delete a database record by selecting option 5 or option 6, respectively. You must know the number of the record you want to update or delete. You can find this by doing a key search or scan. The record will be displayed so that you can verify it. To update a record, simply key in the changes you want to make. Press return to indicate no change. Records are not physically deleted from the database. The location of the delete flag is obtained from the database definition. A # is placed in the delete flag location in the record. This logically deletes the record. A record may be reinstated by using the update mode.

Select option 7 to display the database definition header. Select option 8 to switch to another database.

Other Considerations

DBPROG places certain restrictions on the database. No more than 50 labels are allowed; i.e., the database record must not have more than 50 named fields. This number can be increased by changing the dimensioned sizes of S, P and L$ in line 80. The maximum key length is 60. This is limited by K$. The maximum record length is 500, limited by E$.

DBPROG reads the entire database definition into memory at initialization time. This means that only one physical access is made to this file, even though its individual fields are referenced quite often. The variable L is the next record pointer. Each time a record is added to the data-

Listing 3. Database program DBPROG.

```
10 REM  *****  DATABASE MANAGER  *****
20 REM
30 REM  CODED BY JOHN E. BAILEY
40 REM  SULPHUR, LA   APRIL, 1980
50 REM
60 REM  *****  PROGRAM NAME IS  "DBPROG"  *****
70 REM
80 DIM S(50),P(50),L$(750),K$(60),A$(15),B$(60),E$(500)
90 P1=11
100 FOR I=1 TO P1\!\NEXT
110 !
120 !"     ENTER NAME OF DATABASE (4 CHARACTERS) ? ",CHR$(7),
130 INPUT "",N$
140 IF LEN(N$)<>4 THEN STOP
150 F$=N$+"DEFN,2"
160 OPEN #1,F$
170 READ #1,L,X$,R,K0,K9,D9
180 I=1
190 IF TYP(1)=0 THEN 280
200 READ #1,A$,X,Y
210 J=(I-1)*15+1
220 S(I)=X
230 P(I)=Y
240 K=J+14
250 L$(J,K)=A$
260 I=I+1
270 GOTO 190
280 J=(I-1)*15+1
290 L$(J)="END"
300 T=I-1
310 D$=N$+"DATA,2"
320 OPEN #2,D$
330 FOR I=1 TO P1\!\NEXT
340 F1=0
350 !"                  *****   DATABASE MANAGER   *****"
360 !
370 !"                SELECT ONE OF THE FOLLOWING OPTIONS :"
380 !
390 !"     1 - ADD RECORDS                    5 - UPDATE A RECORD"
400 !"     2 - KEY SEARCH                     6 - DELETE A RECORD"
410 !"     3 - CHARACTER SCAN                 7 - DISPLAY DBDEFN"
420 !"     4 - RECORD NUMBER                  8 - SWITCH DATABASES"
430 !\!
440 INPUT "                         ENTER YOUR SELECTION ? ",N
450 FOR I=1 TO P1\!\NEXT
460 ON N GOTO 510,700,1110,1910,1470,1310,1730,470
470 REM  *****  CLOSE FILES AND RETURN TO START  *****
480 CLOSE #1
490 CLOSE #2
500 GOTO 100
510 REM  *****  ADD RECORDS TO THE DATABASE  *****
520 E$=""
530 FOR M=1 TO R\E$=E$+" "\NEXT
540 FOR Q=1 TO T
550 J=(Q-1)*15+1
560 K=J+14
570 S=S(Q)
580 P=P(Q)
590 IF S=D9 THEN 640
600 !L$(J,K)," ? ",
610 INPUT "",B$
620 IF B$="END" THEN 330
630 E$(S,P)=B$
640 NEXT
650 WRITE #2 %L*(R+2),E$,NOENDMARK
660 L=L+1
670 WRITE #1 %0,L,NOENDMARK
680 !
690 GOTO 510
700 REM  *****  SEARCH DATABASE BY KEY  *****
710 K1=K0
720 INPUT "     ENTER SEARCH KEY ? ",K$
730 IF K$="" THEN K$="ALL"
740 INPUT "     USE ALTERNATE KEY  (Y OR N) ? ",Z$
750 IF Z$<>"Y" THEN 770
760 INPUT "..........ENTER START POSITION OF KEY ? ",K1
770 INPUT "     DO YOU WANT A HARDCOPY (Y OR N) ? ",Z$
```

More

```
                    ENTER YOUR SELECTION ? 2

     ENTER SEARCH KEY ? IC
     USE ALTERNATE KEY (Y OR N) ? N
     DO YOU WANT A HARDCOPY (Y OR N) ? Y


REC. 0
PART NUMBER      :    IC7400
DESCRIPTION      :    QUAD NAND GATE
QTY ON HAND      :    4

REC. 1
PART NUMBER      :    IC7402
DESCRIPTION      :    QUAD NOR GATE
QTY ON HAND      :    5

REC. 4
PART NUMBER      :    IC7408
DESCRIPTION      :    QUAD AND GATE
QTY ON HAND      :    7

REC. 5
PART NUMBER      :    IC7432
DESCRIPTION      :    QUAD OR GATE
QTY ON HAND      :    1

REC. 9
PART NUMBER      :    IC7475
DESCRIPTION      :    QUAD LATCH
QTY ON HAND      :    2

REC. 10
PART NUMBER      :    IC74193
DESCRIPTION      :    SYNC COUNTER
QTY ON HAND      :    1



     ENTER SEARCH KEY ? IC7432
     USE ALTERNATE KEY (Y OR N) ? N
     DO YOU WANT A HARDCOPY (Y OR N) ? Y


REC. 5
PART NUMBER      :    IC7432
DESCRIPTION      :    QUAD OR GATE
QTY ON HAND      :    1
```

Sample run 3. Two examples of retrieving records by key (option 2). The first example shows how each record whose key begins with IC is retrieved when IC is entered for the key. The second example illustrates a method whereby a single record is retrieved when the full key IC7432 is entered.

Listing 3 continued.

```
780 IF Z$="Y" THEN H=1 ELSE H=0
790 FOR I=1 TO P1\!\NEXT
800 M=LEN(K$)-1
810 M=M+K1
820 F=0
830 READ #2 %F*(R+2),E$
840 IF E$(K1,K1)=" " THEN 930
850 IF K$="ALL" THEN 880
860 IF E$(K1,M)=K$ THEN 880
870 GOTO 910
880 GOSUB 970
890 F1=F1+1
900 IF Z$="END" THEN 330
910 F=F+1
920 GOTO 830
930 IF F1>0 THEN 330
940 !"     *****  NO RECORDS FOUND FOR - ",K$
950 INPUT "",Z$
960 GOTO 330
970 REM  *****  PRINT SUBROUTINE  *****
980 IF E$(D9,D9)<>" " THEN RETURN
990 !#H,"REC.",F
1000 FOR Q=1 TO T
1010 J=(Q-1)*15+1
1020 K=J+14
1030 S=S(Q)
1040 P=P(Q)
1050 !#H,L$(J,K)," :   ",E$(S,P)
1060 NEXT
1070 IF H=1 THEN !#H
1080 IF N=5 OR N=6 THEN RETURN
1090 IF H=0 THEN INPUT "",Z$
1100 RETURN
1110 REM  *****  SCAN DATABASE  *****
1120 INPUT "      ENTER SCAN ARGUMENT ? ",K$
1130 INPUT "      DO YOU WANT A HARDCOPY (Y OR N) ? ",Z$
1140 IF Z$="Y" THEN H=1 ELSE H=0
1150 FOR I=1 TO P1\!\NEXT
1160 M=LEN(K$)-1
1170 U=R-M
1180 F=0
1190 READ #2 %F*(R+2),E$
1200 IF E$(K0,K0)=" " THEN 930
1210 FOR I=1 TO U
1220 J=I+M
1230 IF E$(I,J)=K$ THEN 1260
1240 NEXT
1250 GOTO 1290
1260 GOSUB 970
1270 F1=F1+1
1280 IF Z$="END" THEN 330
1290 F=F+1
1300 GOTO 1190
1310 REM  *****  DELETE A RECORD FROM THE DATABASE  *****
1320 !
1330 INPUT "      ENTER RECORD NUMBER ? ",F
1340 IF F<0 THEN 330
1350 IF F>L THEN 1320
1360 READ #2 %F*(R+2),E$
1370 !\!\!
1380 GOSUB 990
1390 !
1400 INPUT "      IS THIS THE CORRECT RECORD TO DELETE (Y OR N) ? ",Z$
1410 IF Z$="Y" THEN 1440
1420 FOR I=1 TO P1\!\NEXT
1430 GOTO 1330
1440 E$(D9,D9)="#"
1450 WRITE #2 %F*(R+2),E$,NOENDMARK
1460 GOTO 330
1470 REM  *****  UPDATE A DATABASE RECORD  *****
1480 INPUT "      ENTER RECORD NUMBER ? ",F
1490 IF F<0 THEN 330
1500 READ #2 %F*(R+2),E$
1510 !\!\!
1520 GOSUB 990
1530 !
1540 INPUT "      IS THIS THE CORRECT RECORD (Y OR N) ? ",Z$
1550 FOR I=1 TO P1\!\NEXT
1560 IF Z$="Y" THEN 1580
1570 GOTO 1480
1580 FOR Q=1 TO T
1590 J=(Q-1)*15+1
1600 K=J+14
1610 S=S(Q)
1620 P=P(Q)
1630 !L$(J,K)," :  ",E$(S,P)
1640 INPUT ".....CHANGE TO  ?  ",B$
1650 IF B$="" THEN 1670
1660 E$(S,P)=B$
1670 NEXT
1680 IF E$(D9,D9)=" " THEN 1710
1690 INPUT ".....RECORD IS DELETED.  RE-INSTATE (Y OR N) ? ",Z$
1700 IF Z$="Y" THEN E$(D9,D9)=" "
1710 WRITE #2 %F*(R+2),E$,NOENDMARK
1720 GOTO 330
1730 REM  *****  DISPLAY DEFINITION HEADER  *****
```

More

base, L is incremented by 1 and rewritten by line 670. This ensures that the last record pointer in the database definition will be correct even if the program abnormally terminates.

A few words about access times. Each access to the database, either for a key search or a scan, requires a sequential read of the entire database. Needless to say, if the record you are searching for is close to the beginning of the database, it will be found sooner than if it is at the end. Search times will be faster for databases with shorter records. I therefore recommend that the data records be kept as short as possible. I have defined one database that has records that are 77 bytes long. To do a key search for record 50 takes about four seconds. Thus, if that database had 200 records in it, the maximum access time should not be greater than 16 seconds.

One other program is presented here. DBVERIFY is used to verify the database (see Listing 4). With it you can list the database definition from the definition file. It also verifies that the record number in the database definition's next record pointer is correct. If there is a difference, the program will notify you and correct the next record pointer. Sample 4 demonstrates the use of the DBVERIFY program.

Conclusion

To recap, follow these steps to define a new database: first, define the data record on paper, its length, the key and its location in the record, each field and its name and location. Then run DBDEFN. Simply answer its questions as per your definition. Next, run DBFRMT to format the database. Then run DBPROG. Select option 1 initially to load records into the database. Then select any of the other options to manage the data as desired.

I designed this database system to allow me to define various data files and access and maintain them with one supervisory program. The system sacrifices access time to allow duplicate keys and to eliminate the need for an ordered database. Even with its limitations, the program is excellent for cataloging many kinds of data, such as parts lists, recipe lists, personal checks, name and address lists, record album catalogs and household inventories. I've found that its capabilities far outweigh its limitations.■

Listing 3 continued.

```
1740 !\!
1750 !TAB(10),"NAME OF DATABASE       -  ",X$
1760 !TAB(10),"NO. OF REC. IN FILE   - ",L
1770 !TAB(10),"RECORD LENGTH         - ",R
1780 !TAB(10),"KEY START POSITION    - ",K0
1790 !TAB(10),"KEY STOP POSITION     - ",K9
1800 !TAB(10),"POS. OF DEL. FLAG     - ",D9
1810 FOR I=1 TO T
1820 IF S(I)=K0 THEN 1860
1830 NEXT
1840 !
1850 GOTO 1880
1860 I=(I-1)*15+1
1870 !\!"            RECORD KEY IS : ",L$(I,I+14)
1880 !\!
1890 INPUT "",Z$
1900 GOTO 350
1910 REM  *****  RETRIEVE A RECORD BY RECORD NUMBER
1920 FOR I=1 TO P1\!\NEXT
1930 INPUT "      ENTER RECORD NUMBER ? ",F
1940 IF F<0 OR F>L-1 THEN 1920
1950 !\!\!
1960 READ #2 %F*(R+2),E$
1970 GOSUB 970
1980 GOTO 330
1990 END
```

```
*****  VERIFY DATABASE  *****

ENTER NAME OF DATABASE ? PRTS

DO YOU WANT A HARDCOPY (Y OR N) ? Y


DATABASE NAME         -  PRTS
NO. OF REC. IN FILE   -  12
RECORD LENGTH         -  28
KEY START POSITION    -  1
KEY STOP POSITION     -  10
POS. OF DEL. FLAG     -  28

LABEL               START  STOP
PART NUMBER     :      1     10
DESCRIPTION     :     11     25
QTY ON HAND     :     26     27


*****  END OF DATABASE VERIFY  *****
```

Sample run 4. DBVERIFY is used to verify the database. It lists the database definition and verifies that the next record pointer is correct.


```
10 REM  *****  VERIFY DATABASE  *****
20 REM
30 REM  CODED BY JOHN E. BAILEY
40 REM  SULPHUR, LA   APRIL, 1980
50 REM
60 REM  *****  PROGRAM NAME IS  "DBVERIFY"  *****
70 REM
80 DIM A$(15)
90 P1=11
100 FOR I=1 TO P1\!\NEXT
110 !"      *****  VERIFY DATABASE  *****"
120 !
130 INPUT "      ENTER NAME OF DATABASE ? ",N$
140 IF LEN(N$)<>4 THEN 120
150 F$=N$+"DEFN,2"
160 OPEN #1,F$
170 !\INPUT "      DO YOU WANT A HARDCOPY (Y OR N) ? ",Z$
180 IF Z$="Y" THEN H=1 ELSE H=0
190 FOR I=1 TO P1\!\NEXT
200 READ #1,L,X$,R,K0,K9,D9
210 !#H,"      DATABASE NAME         -  ",X$
220 !#H,"      NO. OF REC. IN FILE  - ",L
230 !#H,"      RECORD LENGTH        - ",R
240 !#H,"      KEY START POSITION   - ",K0
250 !#H,"      KEY STOP POSITION    - ",K9
260 !#H,"      POS. OF DEL. FLAG    - ",D9
270 !#H
280 !#H,"      LABEL                START  STOP"
290 IF TYP(1)=0 THEN 340
300 READ #1,A$,S,P
310 !#H,"      ",A$," :  ",%4I,S,"  ",P,
320 IF S=K0 THEN !#H"   - KEY" ELSE '#H""
330 GOTO 290
340 IF H=0 THEN INPUT "",Z$
350 D$=N$+"DATA,2"
360 DIM E$(R)
370 OPEN #2,D$
380 I=0
390 READ #2,E$
400 IF E$(K0,K0)=" " THEN 430
410 I=I+1
420 GOTO 390
430 IF I<0 THEN I=0
440 IF I=L THEN 540
450 !#H\!#H
460 !#H,"      *****  RECORD COUNTS DO NOT MATCH  *****"
470 !#H
480 !#H,"      NO. OF RECORDS IN DEFN  = ",L
490 !#H,"      NO. OF RECORDS IN DATA  = ",I
500 !\!\INPUT "      TYPE STOP TO AVOID CORRECTING DBDEFN ? ",Z$
510 IF Z$="STOP" THEN 540
520 WRITE #1 %0,I,NOENDMARK
530 !#H\!#H"..........DATABASE DEFN UPDATED."
540 !#H\!#H
550 !#H,"      *****  END OF DATABASE VERIFY  *****"
560 CLOSE #1
570 CLOSE #2
580 !#H\!#H
590 END
```

Listing 4. Database verification program DBVERIFY.

✓ Reader Service—see page 178

Thoughts on the 68XX System

By Peter A. Stark

The suggestion has been made that I rename this series, as it really covers much more than just the SWTP computer system. Thus the new name, which more adequately describes the fact that most of the material in these columns applies not just to SWTP mainframes, but to 6800, 6802 and 6809 systems made by others as well.

Real-Time Clock

January's installment described a status line on a video board display, and mentioned several uses for such a line. Let's follow that up a bit more.

There are various status line displays which do not require a real-time clock—for instance, a disk operating system could be easily patched to display the drive, track and sector numbers in a particular spot on the video screen each time it reads or writes a sector.

On the other hand, a real-time clock simplifies the display of other status information. For that reason we should examine what it is.

There is a difference between a clock and a real-time clock. A clock shows the actual time; a real-time clock only indicates passing time. A real-time clock can be compared with a metronome—it ticks like a clock, but it doesn't have a display to show the current time. However, we can figure out how much time has passed from any beginning point by counting how many ticks we have heard.

An actual clock board is quite a bit more complicated than a real-time clock. Several clock boards are available (one is the Microtime board available from several dealers; others are available from JPC Products, PO Box 5615, Albuquerque, NM 87185, and Robertson Electronics, 1003 Warm Sands Drive, SE, Albuquerque, NM 87123). A calendar clock circuit for the 6800 was described in the June 1980 *Kilobaud Microcomputing* (p. 188).

Each of these circuits uses a series of counters like those in a digital watch, which count up by one each second and provide a constant readout of the current time. In order to avoid losing time when the power is off, most such circuits allow a battery to power the counters so they can continue to run when the rest of the computer is shut down. Once such a clock is set, it continues to indicate the actual time in hours, minutes and seconds (and sometimes the date as well).

A real-time clock, on the other hand, does not indicate the time directly. Instead, it provides ticks which can be used to compute the time if desired (assuming that the time between ticks is precisely known and controlled).

These ticks generally cause IRQ interrupts, which cause the 6800 to temporarily suspend normal programming and jump to an interrupt service subroutine (ISS).

When such an interrupt arrives, it can be used to compute current time, display some system variables or switch between tasks in a multiprogramming system (which is a concept we discussed in the October 1980 installment).

A real-time clock is more than just an oscillator which pulses the IRQ line. It does do that, of course, but it also has circuitry which can turn the clock on and off, so that it does not start to generate clock pulses until the system has an ISS in place and ready. Moreover, since the 6800 might interrupt more than once if the clock pulse were wide enough, the clock board must have circuitry which provides a pulse long enough to be detected even if the 6800 is busy with another job and has its interrupt system temporarily disabled, but turns off the IRQ pulse as soon as the 6800 is actually interrupted. Thus, a real-time clock usually consists of several ICs.

SWTP makes a real-time clock called an MP-T timer, which is programmable and can cause interrupts (ticks) to occur at a wide range of precisely controlled intervals. However, if you want to experiment with a real-time clock, but don't need such precise timing, you can use a spare MP-S serial interface board. (Make sure that it has an IRQ jumper installed.)

When a character is sent to the ACIA in the MP-S by the 6800 proces-

Peter A. Stark, PO Box 209, Mt. Kisco, NY 10549.

sor, the ACIA sends it out in serial—one bit at a time—at the baud rate selected by the jumpers on the board. When it has finished sending the character, it turns on the TDRE (transmitter data register empty) bit in its status register. This is how software generally knows when it is time to send the next character. If the MP-S is set to 300 baud, for example, then the TDRE bit will come on 1/30 second after the ACIA receives a character from the 6800.

If, however, the ACIA is properly initialized, it can also generate an interrupt at the same time that TDRE goes on. Consequently, we can use an ACIA to generate a fixed time delay, and then cause an interrupt. If we restart the ACIA as soon as we receive that interrupt, then it will start all over again. Thus, it can provide a constant series of ticks. It need not actually have any I/O device connected to it to do this; in fact, it's better if it is an extra MP-S board in some otherwise unused slot.

Depending on how the ACIA is initialized, we do have some control over the time interval between ticks. This time depends on the baud rate, the number of bits in each character, and the clock divisor used by the ACIA. We'll leave the fine points to you, and consider just a simple case.

Let's assume that we have an MP-S card in port 0; the ACIA uses address 8000 for its control and status register, and address 8001 for data. To initialize the ACIA to turn its interrupts on, we place $35 into the control register (location 8000); to turn interrupts off, we replace it with $15.

If we turn on the ACIA's interrupt system and then start the ACIA by placing a data character into its data register, location 8001, it will output that character in serial. When it is finished, it turns on the TDRE bit in the status register (location 8000), sends out a low pulse on the IRQ line to the 6800 and turns on the IRQ bit, bit 7, in the status register (location 8000). A program can test this bit to check whether the interrupt came from the ACIA or somewhere else.

Listing 1 shows an actual program which makes use of this idea. The program consists of two parts—an initialization routine which starts the ACIA going, and an ISS which is called at each tick, and which (among other things) restarts the ACIA.

At location B000, the program resets the ACIA by storing 03 into it; the ACIA does not have a reset pin and so must get reset by software when first starting up the system.

At location B005, the program stores the address of ISS into address A000, which is the monitor's IRQ vector. This means that when an IRQ occurs, the monitor will steer the computer to execute the ISS program, and thus start the interrupt service subroutine.

After some more initializing (more on that later), at location B01E we store a $35 into the ACIA's control register to enable its interrupts. This is followed by sending out character $FF to be printed. Although any character could have been chosen, the $FF is a rubout which will not be printed even if you have a printer connected to the ACIA. (However, a word of caution—although the ACIA can have an I/O device connected to it, that device must not be used while we are using the MP-S port for a real-time clock!)

Finally, a CLI instruction enables the interrupt system in the 6800 (it is preceded by a NOP, since some earlier 6800 ICs did not work properly unless CLI and SEI instructions were preceded by NOPs).

At this point, we assume that the initialize routine returns to whatever program called it, and this program continues.

When the ACIA has finished outputting, however, it generates an interrupt. As a result, the above program is interrupted, and the computer comes to the interrupt service subroutine at the entry point ISS.

Since interrupts can come from several places in the computer, we begin by testing whether it in fact came from where we think it came from. This is done at location B02B, where we load the status bits from the ACIA, move the leftmost bit (the IRQ bit, bit 7) into the carry bit and continue to OK if the carry is set.

If the carry is not set, the interrupt came from elsewhere. In that case,

Listing 1. MSCAN program.

```
                       NAM    MSCAN

                  * THIS PROGRAM PRODUCES A MEMORY USAGE MAP
                  * IN THE VIDEO STATUS LINE

                  * EXTERNAL REFERENCES
 (A000)           IRQ    EQU   $A000      IRQ VECTOR
 (8000)           ACIA   EQU   $8000      PORT 0 ACIA
 (E07E)           PDATA  EQU   $E07E      PRINT STRING ROUTINE
 (E0E3)           MONITR EQU   $E0E3      MONITOR RETURN POINT
 (D800)           SCRAM  EQU   $D800      SCREEN RAM START
 (D010)           TOP    EQU   $D010      USABLE RAM TOP ADDRESS

                  * INITIALIZE AND START DISPLAY ROUTINE

 (B000)                  ORG   $B000
B000 86 03        INITLZ LDA A #$03
B002 B7 8000             STA A ACIA       RESET ACIA BEFORE STARTING
B005 CE B02B             LDX   #ISS
B008 FF A000             STX   IRQ        INITIALIZE INTERRUPT POINTER
B00B CE D800             LDX   #SCRAM     POINT TO START OF SCREEN RAM
B00E FF B073             STX   PREV       INIT PREVIOUS POINTER
B011 86 20               LDA A #$20       GET A SPACE
B013 A7 00        ERASE  STA A 0,X        ERASE NEXT BYTE
B015 08                  INX
B016 8C D850             CPX   #$D850     END OF LINE?
B019 26 F8               BNE   ERASE      NO
B01B FF D010             STX   TOP        YES; RESERVE A LINE
B01E 86 35               LDA A #$35       INIT ACIA CR TO 00110101
B020 B7 8000             STA A ACIA
B023 86 FF               LDA A #$FF
B025 B7 8001             STA A ACIA+1     START ACIA ACTION
B028 01                  NOP
B029 0E                  CLI              ENABLE INTERRUPTS
B02A 39                  RTS

                  * INTERRUPT SERVICE SUBROUTINE

B02B B6 8000      ISS    LDA A ACIA       TEST FOR INTERRUPT
B02E 48                  ASL A
B02F 25 19               BCS   OK         IF CAUSED BY TIMER
```

More →

Listing 1 continued.

```
B031 86 15             LDA A #$15       TURN OFF ACIA INTERRUPT
B033 B7 8000           STA A ACIA
B036 CE B03F           LDX   #ERRMSG    OTHERWISE PRINT ERROR MESSAGE
B039 BD E07E           JSR   PDATA
B03C 7E E0E3           JMP   MONITR
B03F 41         ERRMSG FCC   'ACIA ERROR'
B049 04                FCB   4

B04A 30         OK     TSX              TRANSFER STACK PTR TO INDEX
B04B E6 05             LDA B 5,X        GET PC HI
B04D 54                LSR B
B04E 54                LSR B            MOVE MSB TO RIGHT
B04F 54                LSR B
B050 54                LSR B
B051 F1 B074           CMP B PREV+1
B054 27 17             BEQ   SAME       SKIP IF SAME AS PREV
B056 FE B073           LDX   PREV       POINT TO PREV DIGIT
B059 86 20             LDA A #$20
B05B A7 00             STA A 0,X        ERASE IT
B05D F7 B074           STA B PREV+1     POINT TO NEW LOCATION
B060 CB 30             ADD B #$30       CHANGE DIGIT TO ASCII
B062 C1 3A             CMP B #$3A       IS IT 0-9?
B064 25 02             BCS   NUMBER     YES
B066 CB 07             ADD B #$07       NO, CHANGE TO A-F
B068 FE B073    NUMBER LDX   PREV       POINT TO NEW LOCATION
B06B E7 00             STA B 0,X        MODIFY IT
B06D 86 FF      SAME   LDA A #$FF       RESTART ACIA
B06F B7 8001           STA A ACIA+1
B072 3B                RTI              AND RETURN

B073            PREV   RMB   2          POINTER TO PREVIOUS LOCATION

                       END
```


we turn off the ACIA interrupt system by putting $15 into it (at location B031), print an error message and jump to the monitor.

If the interrupt really came from our real-time clock, then we continue at OK. After some more processing, we restart the ACIA by sending another $FF to it in location B06D, and execute an RTI, or return from interrupt. This returns to the main program, which will execute until the next interrupt tick comes. In this case, we get about 30 interrupts per second if the ACIA is set for 300 baud.

This explains how we can use an extra MP-S interface's ACIA as a simple real-time clock. Now what can we do with it?

Program MSCAN

January's article showed how to reserve an extra status line on a Percom video board display. The same idea can be applied to any of the many other video boards available. Listing 1, called MSCAN, is a program which makes use of this to produce a memory usage map as other programs are running.

This is a display in the top left corner of the screen which indicates what addresses are currently being used by the running program. When a program which is located in the range of 1000–1FFF is running, for example, the screen shows the display

.1..............

(except that the screen has blank spaces rather than periods). If a program located up in the 6000 area is running, then the display is

......6.........

If the programs were rapidly jumping back and forth to each other, then the display would be

.1....6.........

with the 1 and 6 flashing alternately. By observing this kind of display, we can get an idea whether programs are running correctly.

In this particular version, we assume that the screen RAM starts at location D800 (which is called SCRAM in the program) and that the top line is reserved for status. Locations D800–D84F are used for status, and the regular display starts at D850. The video driver we described in January requires that location D010 be set to D850 to achieve this.

With this as a background, let's see how the program works.

During initialization, the SCRAM address of D800 is placed into location PREV, which is used to indicate the previous digit displayed in the map.

The next part of the program, at location B011–B019, erases the 80-byte top line of the display, and then the number D850 is placed into TOP to tell the video driver not to use the top line. All of this is, of course, sandwiched between the ACIA initialization we discussed earlier.

Each time an interrupt occurs, the computer goes to the ISS routine. After some preliminaries, it reaches OK at location B04A, where the TSX instruction moves the contents of the stack pointer (plus 1) into the index register. Five bytes up is the first half of the program counter, which tells us the address of the instruction which would have been performed next had there been no interrupt. This is loaded into the B register.

A series of LSR (logical shift right) instructions moves the leftmost hex

Reader Service—see page 178

digit to the right, and it is now compared with the right byte of PREV. Because the screen memory starts at an even boundary of D800, if the PC started with 0 we want to display 0 at location D800, and so on, up to F at location D80F if the PC started with F.

Since PREV points to the previous digit's location, we can look at its rightmost byte to determine what digit is currently being displayed. If it is still the same, the program simply goes to SAME, resets the ACIA and exits. By not updating the screen when the display is not changing, we avoid unnecessary flicker.

If the new PC address starts with a different digit from the previous one, then we update the display. First, at B056 we point the index register to the location of the previous digit and then replace it with a space to erase it. Then we change PREV by storing the leftmost digit of the PC (which has been in the B accumulator all this time) into its second byte. The next few instructions convert this digit into ASCII. Finally, at location B068, we point the index register to the new location (which is now in PREV) and store the new ASCII character in that spot. Then we restart the ACIA and return from the interrupt to the main program.

Variations on this program can be used to display other useful facts in the status line of the display; just add the additional steps where SAME is.

Even if keeping a status line is of no interest to you, if you have a video board then this is a simple program which can give you a little experience using interrupts.

Programming the 2732 EPROM

The 2732 4K×8 EPROM (which is not the same as TI's TMS-2532) is sufficiently similar to the 2716 that it can be programmed with the SWTP MP-R programmer. Only three wiring changes and some program changes are required. (I covered 2708 programming on p. 82 of the February 1980 issue of *Kilobaud Microcomputing*.)

The 2732 pin functions differ from the 2716 in several ways. First, pin 21 (which is called Vpp on the 2716 and gets +25 V during programming) now gets A11, an additional address bit. To implement this change, cut the copper trace which goes from diode D1 to pin 21 of the EPROM socket and also cut the trace connecting R7 to pin 12 of IC3. Then connect EPROM pin 21 to IC3, pin 12, and connect the cut lead of R7 to IC3, pin 9. (This changes the control bit for the +5 V supply, but that is taken care of by software.)

Next, pin 20 of a 2716 is an active low chip select, whereas pin 20 of a 2732 is a combination $\overline{OE}$/Vpp pin which is to be low normally, but at +25 V during programming. To implement this change cut the trace connecting pin 20 of the EPROM socket to IC3, pin 5, and connect pin 20 instead to the anode side of diode D2.

Next, make sure that a 0.1 uF disk capacitor has been installed in parallel with R6, as instructed in an addendum to the SWTP instruction sheets.

One more caution is in order. The Intel 2732 EPROM requires +25 V for Vpp, which is supplied by the MP-R board. However, the Intel 2732A EPROM (note the A after the 2732!) requires only 21 V and can be damaged if more than 21.5 V is applied. If you are using the 2732A, you

must include some additional regulation—perhaps using a Zener diode—to limit Vpp.

Next, make the changes shown in Listing 2 to the SWTP program. These change the buffer size and the sequencing of the power to supplies and programming pulses.

Finally, save the modified program from 0020 through 07E1, with a starting address of 0100.

Video Boards as Terminals

One of the unexpected expenses in setting up an SS-50 bus system is the need for a terminal. This is one of the factors which often swing a newcomer in the direction of a Radio Shack or Apple system, since the price of an SS-50 system plus a terminal usually exceeds the price of an all-in-one system.

Needless to say, a serial terminal is not absolutely needed; a video board and parallel keyboard can do the job as well, and at much lower cost.

Making the video board and the keyboard work with standard programs requires two separate interfaces—one software, the other hardware.

```
0037  0FFF          CHANGE BUFFER END ADDRESS
0422  86 10         TURN ON ONLY +5 VOLTS
044D  86 10         TURN ON ONLY +5 VOLTS
0479  86 10         TURN ON ONLY +5 VOLTS
04A8  BD 07C0       JSR TO PATCH 1
04AB  01
04C6  BD 07D0       JSR TO PATCH 2
04C9  01
0561  86 90         +5V ON, +25V ON, CE' PULSED LOW
0573  86 B0         +5V ON, +25V ON, CE' BACK TO HI
0580  86 30         +5V ON, +25V OFF, CE' KEPT HI
* PATCH 1 - TURN ON +25 VOLTS BEFORE PROGRAMMING
07C0  86 20         +5V ON, CE' HI
07C2  97 26         STORE IT
07C4  BD 05D6       JSR TO PIA PULSING ROUTINE
07C7  86 B0         +5V ON, +25V ON, CE' STILL HI
07C9  97 26
07CB  BD 05D6       JSR TO PIA PULSING ROUTINE
07CE  39            AND RETURN
* PATCH 2 - TURN OFF 25 VOLTS AT END OF PROGRAMMING
07D0  86 30  +5V ON, +25V OFF, CE' HI
07D2  97 26
07D4  BD 05D6       JSR TO PIA PULSING ROUTINE
07D7  BD 04FE       WAIT 10 SECONDS
07DA  86 10         +5V ON, +25V OFF, CE' BACK TO LOW
07DC  97 26
07DE  BD 05D6        JSR TO PIA PULSING ROUTINE
07E1  39             AND RETURN
```

Listing 2. SWTP programmer modifications.

For terminal output, standard 68xx software simply outputs via a serial board in port 1. Although older systems use a PIA (in a circuit which is a

relic of MIKBUG days), the modern approach is to use an ACIA circuit such as that of the SWTP MP-S serial card. The terminal output routine—part of the ROM monitor and usually called OUTEEE—consists of three essential steps (two to check whether the ACIA is busy, and one to output the actual character) plus a few extra instructions which pick up the port address and configure the ACIA for proper output. The rest of the terminal functions—scrolling, moving the cursor, etc.—are taken care of by the terminal itself.

When using a video board, these ancillary functions must be taken care of by software. As described in January's article, scrolling is sometimes partially handled by video board hardware, and sometimes completely done by software. In either case, however, an extensive amount of software is needed to do the job.

When a separate terminal is used in addition to the video board, it is feasible to keep this software on tape or disk and load it into the computer's RAM each time the computer is turned on. But when no terminal is present, then the video board software must be located in ROM; moreover, it must be linked to the monitor program so that it is automatically brought up as power is turned on. In other words, a special monitor program must be used with the video board in that case.

Fortunately, a monitor does exist for every current video board being made—Gimix has GMXBUG, Thomas has JOEBUG, and Percom's video board is supported by my HUMBUG monitor (described in the August 1980 installment and available from Star-Kits). Using a video board, then, simply involves getting the appropriate monitor ROM to handle it.

Parallel keyboard input, on the other hand, is not so easy. The standard procedure is to connect the keyboard through a parallel interface, such as the SWTP MP-LA card. The seven-bit data from the keyboard goes into the *data port*, while the data strobe bit goes into one of the PIA handshaking lines on the port. Since the PIA is intended for just such parallel inputs, this is a good hardware match. (On disk systems this PIA interface would be used instead of an ACIA interface; on tape systems both may be required—one for the keyboard and the other for the tape interface.)

But there is usually some software required as well. Since the routine needed to initialize and input from a PIA differs from that needed for an ACIA, the INCH or INEEE routine must also be different.

As before, if a serial terminal is available, then this revised routine could be on tape or disk and entered into RAM at power-up. But lacking a terminal, the routine must also be part of the monitor in ROM so that it is operational as soon as power comes on. Some of the video-based monitors have it; some don't. But there is more to it than that.

Much user software—like BASIC interpreters, editors, disk operating systems, assemblers, etc.—monitors the keyboard, waiting for a break character such as a control-C. This function is usually performed directly by each program, rather than going through the monitor.

When a PIA is used with a parallel keyboard, each such program must be patched to monitor the PIA rather than the standard ACIA. Not only is this a tedious job (for example, the instruction manual for the Gimix video board has 15 pages of patches and does not even cover all the possible programs), but it also means that the patched software is nonstandard and may not run on other systems. This is a definite liability, since one of the prime differences between SS-50 bus systems and S-100 systems has traditionally been the compatibility between different systems.

There is, of course, a solution. Don't use a PIA at all, but interface the parallel keyboard through an ACIA. This makes the interface truly transparent to all software and avoids

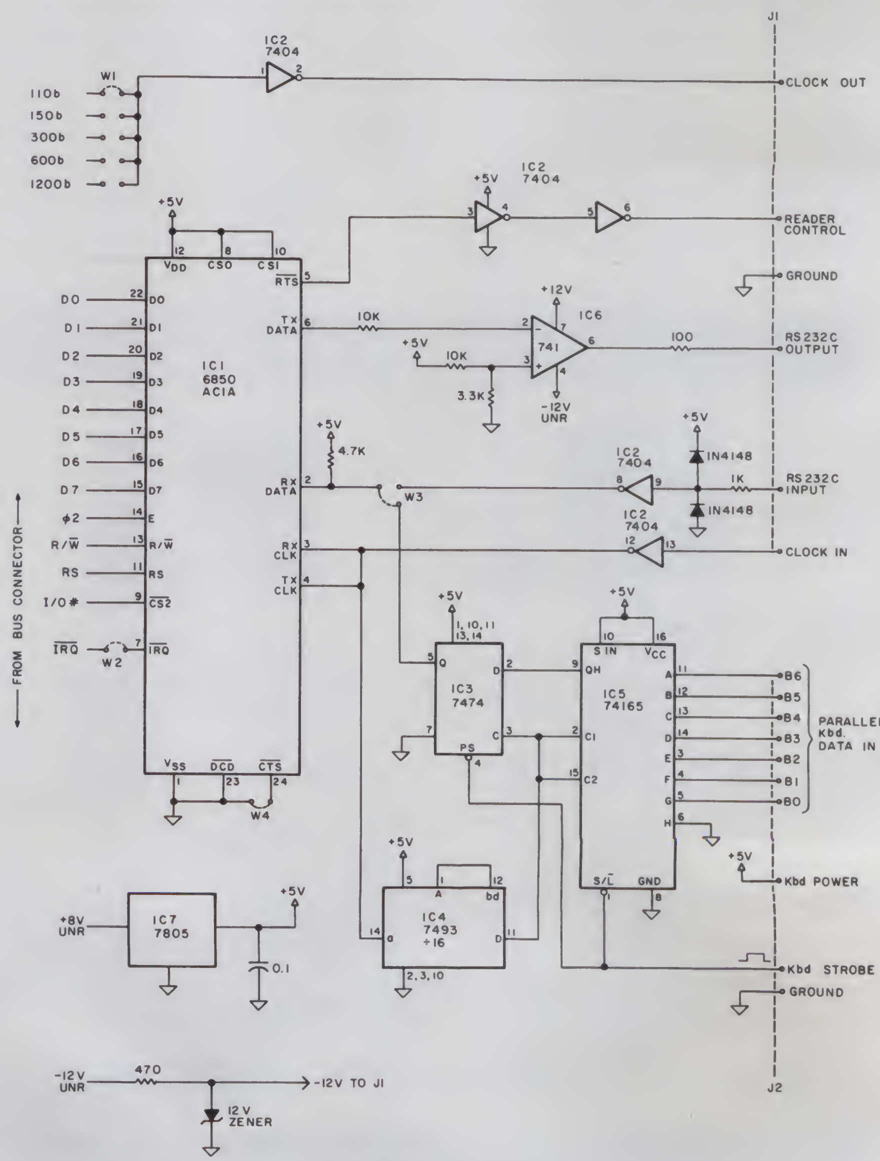


Fig. 1. Serial/parallel interface design.

AT LAST!

Mass production prices for high quality software. Buy direct and save 50%. Also available for CPM and HDOS.

DATA BASE MANAGER Mod I & III $69. $149 (48K). Mod-II $199
Maintain a data base and produce reports, all without user programming. Define file parameters and report formats on-line. Key random access, fast multi-key sort, field arithmetics, audit log, label. No time-consuming overlays. 500 happy users in one year. Mod-II and 48K versions have over 50 enhancements, including 40 fields maximum. "IDM-M2 is great!" - 80-US.

A/R Mod-I $69 Mod-II $149 Mod-III $69
Handles invoices, statements, aging, sales analysis, credit checking, forms input, and order entry. Unlike other accounts receivable programs, ours can be used by doctors, store managers, etc.

WORD PROCESSOR $49
Centers, justifies, indents, and numbers pages. Mod-I version features upper/lower case without hardware modification! File merge option available.

MAILING LIST Mod I & III $59. $79 (48K). Mod-II $99
The best! Compare and be selective. Includes forms input, 5-digit selection code, zip code extension, sort on any field, and multiple labels. Who else offers a report writer and merges with word processor?

INVENTORY Mod I & III $89. $109 (48K) Mod-II $149
Fast key random access. Reports include order info, performance summary, EOQ and user-specified reports. Many people have converted to our system! Next to impossible to damage the file.

GL. A/R. A/P. PAYROLL Mod-II $129 each
Integrated accounting package. 100+ page manual. As opposed to Osborne's slow binary search and 64 column screen, we use fast ISAM and 80 columns. Dual disk and TRSDOS required.

L216 $59
A cassette package of 10 business programs for Level II 16 K systems. Includes word processor and data base manager. Poker game $19

Most programs are on-line, interactive, random-access, bug-free, documented, and delivered on disks. Mod-I programs require 32K TRSDOS. We're #1 in business software — don't let our low price fool you! Ask for our free 20-page catalog if you're still not convinced. Compiled versions are available.

MICRO ARCHITECT, INC.
96 Dothan St., Arlington, MA 02174 ✓108

LEARN MICROPROCESSING ON YOUR OWN COMPUTER!

You Learn How To:

- Design and code microprocessor software
- Use logic and Bit Manipulation Techniques
- Enter and execute programs on your own computer
- Understand Microprocessor Architecture and Support Chips
- Control Programmable Input/Output Ports
- Implement Real-Time Interrupt Handling and Data Transfer
- Design your own microcomputer

8085 MICROPROCESSOR TRAINING UNIT $299.95

RATED BEST VALUE BY INSTRUCTORS

You Receive:

- A fully tested and assembled 8085A Microcomputer with 1K RAM, 1K EPROM and 1k PROM Memory, Programmable I/O, Keyboard Unit, CPU Card, Display and Operating System, 44 pin edge connector can be configured to any bus structure, area on CPU Card for Custom wire-wrap design or user defined interface circuitry, completely expandable
- Complete Step-by-Step Instruction Manual
- Complete User's Manual with programs included
- 352 page 8085A Cookbook takes you from basic microprocessor concepts to actual design of an 8085A Microcomputer
- 344 page 8080/8085 Software Design Book 1 with over 190 executable program examples plus detailed examination of all 244 Instructions and typical assembly language program for the 8080/8085 Microprocessor.

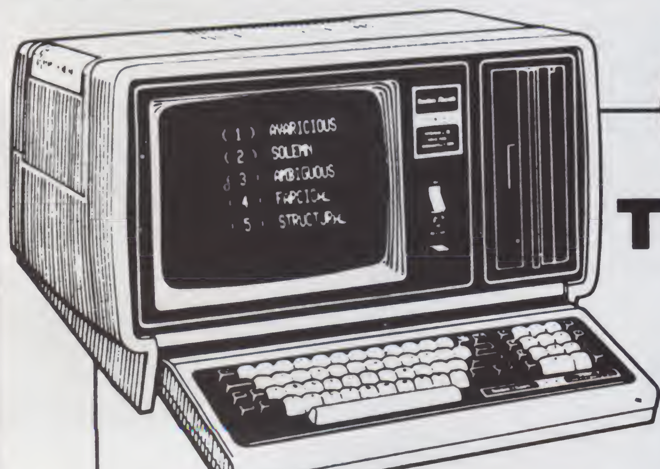
Satisfaction Guaranteed. If not completely satisfied you may return the product within 30 days for a full refund

PACCOM
14905 NE 40th, Dept. KB 3
REDMOND, WA 98052 ✓19

FOR BANK CARD ORDERS:
CALL TOLL-FREE NUMBER:
1-800-426-6254 EXT 93

YES! I want to start learning Microprocessors. Please rush me:
☐ **8085AAT Microprocessor Training Unit at $299.95** plus $3.00 P & H
☐ 8085 Cookbook **21697 $12.95** ☐ 8080/8085 Software Book **21541 $9.50**

NAME	CARD NO.
ADDRESS	EXP. DATE
CITY	☐ VISA ☐ MASTERCARD
STATE ZIP	SIGNATURE

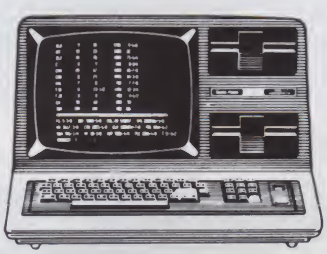


Radio Shack® DEALER

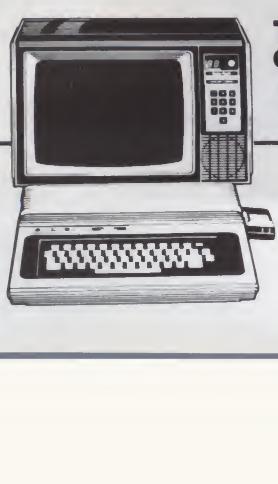
TRS-80™ DISCOUNT

☐ NO OUT-OF-STATE TAX
☐ NO SHIPPING COSTS

TRS-80 MODEL II 64K $3500
PACKS ENOUGH DATA HANDLING POWER FOR MANY SMALL BUSINESSES.



TRS-80 MODEL III 32K-2 DISKS
$2100
NEW PERSONAL COMPUTER . . . REAL-TIME CLOCK, *SHARPER* CRT IMAGES AND FASTER LOADING CASSETTES



TRS-80 COLOR COMPUTER OR VIDEO $339 each
A LOW COST, COLOR COMPUTER FOR PERSONAL BUSINESS OR ENTERTAINMENT

CERTIFIED CHECKS
CASHIERS CHECKS
OR CREDIT CARDS

PERRY OIL & GAS INC. ✓266

137 NORTH MAIN STREET, PERRY, MICH. 48872
PHONE (517) 625-4161

WARRANTIES HONORED BY ALL RADIO SHACKS • *T.M. TANDY CORP.

all incompatibilities. (And, in those systems which use interrupts for keyboard input—such as some of the sophisticated disk operating systems or multitasking systems—this approach is almost mandatory.)

On the surface that sounds impossible, but actually all it requires is that the parallel output of the keyboard be converted to serial, and that this serial signal then be fed to the ACIA as if it came from a terminal.

One approach is to put a parallel-to-serial converter right into the keyboard enclosure and use serial output directly. George Risk Industries makes such an adapter for their GRI keyboards, and similar adapters are also available from Electronic Systems and elsewhere. Though this simplifies the wiring between the keyboard and computer, it requires a more complicated power supply in the keyboard. Moreover, it is not necessarily an economical approach.

The second approach is to combine the serializer with the ACIA port together on one 30-pin interface card. Since the ACIA (and a UART, if used) would require TTL buffering of their I/O signals anyway, combining the parallel-to-serial conversion function on the same circuit board can solve the problem without any increase in circuitry by combining buffering and data conversion in the same set of ICs.

An excellent circuit for a serial/parallel interface board is shown in Fig. 1. This circuit provides not just a parallel keyboard input, but also a full RS-232C I/O port; either of these may be used individually, or both may be used together.

As Fig. 1 shows, a 6850 ACIA is used as in most other serial interfaces. It connects directly to the 30-pin I/O bus of the computer (with jumper W2 being used only when interrupt operation is desired).

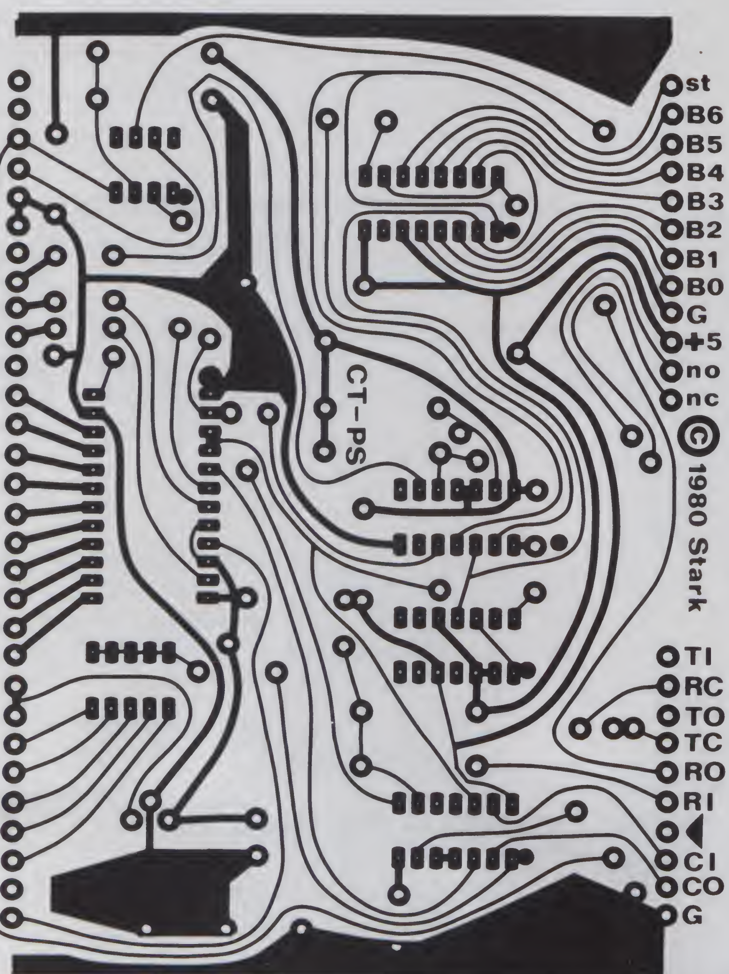


Fig. 2. Board layout.

Jumper W1 is used to select one of the baud rates; the desired baud rate clock is then buffered via a 7404 and sent out to the RS-232C interface connector. In the printed circuit board layout in Fig. 2, this connector is pin-for-pin compatible with that used on the SWTP MP-S card, and the clock is brought out so it may be used for terminal baud rate switching or for tape interface operation.

The clock input signal is again buffered and sent to pins 3 and 4 of the ACIA. In normal operation, the clock out and the clock in pins on the interface connector should be jumpered together. The buffering is included to eliminate data errors often present when an SWTP MP-S interface is separated from a tape interface by more than a foot or two of cable.

Serial RS-232 data is buffered with a 741 op-amp for output, and with another section of a 7404 inverter for input. The reader control line is also buffered by two inverters. Since this interface does not provide current-loop operation, this circuitry is much simpler than on other serial interfaces.

The major difference is in the circuit at the bottom. The seven bits of keyboard data (no parity bit is used) are sent to a 74165 shift register. This IC has eight inputs, one of which (pin 6) is permanently grounded to provide the 0 start pulse which always precedes a serial character.

The output of the shift register, on pin 9, goes to IC3, part of a 7474 type D flip-flop, which is normally kept set by a low on the PS (pre-set) input. The output of this flip-flop on pin 5 is thus normally high (representing a 1) between characters.

Since the ACIA requires a baud rate clock 16 times the actual baud rate, whereas the 74165 shift register requires a clock at the same baud rate, a 7493 divider divides the clock by 16.

The keyboard strobe must be normally low, with a high pulse extending over an entire one-character interval; that is, if the ACIA runs at 300 baud, then the strobe pulse must last for at least 1/30 second; at 1200 baud, it must last at least 1/120 second. This strobe pulse is normally low, which keeps the 74165 from shifting and keeps the 7474 set; when a character is received it goes high, which allows the 74165 and 7474 ICs to pass the data to the ACIA. (The serial input of the 74165, pin 10, is connected to +5 V so that the shift register fills up

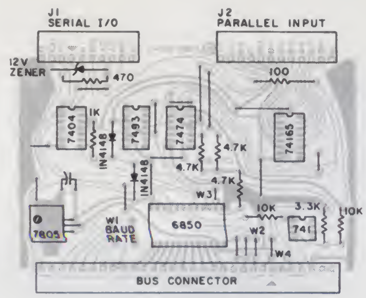


Fig. 3. Parts layout.

with ones as it shifts, and the character is automatically followed by a one stop bit.)

The position of jumper W3 depends on the input data source—it can connect the ACIA to either the 7404 in the serial terminal input or to the 7474 in the keyboard input circuit. If both inputs are to be used simultaneously, the jumper should be replaced by two diodes—both anodes go to ACIA, pin 2, with one cathode going to the 7404 output and the other to the 7474 output.

Power for the board is regulated by a 7805 three-terminal regulator; the 12 V zener diode and its resistor are required only when an I/O device (such as the Percom CIS-30 cassette interface) requires −12 V.

This circuit can be wired on a 30-pin prototyping card, or you can use the printed circuit board layout in Fig. 2; see Fig. 3 for the parts layout. A single-sided board makes construction easy. (If you'd rather obtain a commercial board than make your own, it is available from Star-Kits, PO Box 209, Mt. Kisco, NY 10549, for $20 for a bare board, or $55 for a complete kit of parts.)

The printed circuit board layout also has provisions for debouncing an abort switch (see January's installment). This circuit uses part of the 7474 flip-flop IC (Fig. 4).

A 6800 Cross Assembler

If you have switched to a 6809, but would like an inexpensive 6800 assembler that will run on your system (or if you just want to see how a 6800 assembler works) look at the Kilobaud Klassroom No. 23 article in the December 1980 issue of *Kilobaud Microcomputing*. It describes a 6800/6802 cross assembler written in BASIC. It runs on almost any system which supports a BASIC with strings. I've run it on a TRS-80 with very good results.

The Heath ET-3400 Trainer

Perhaps the most popular 6800-based computer ever made is the Heath ET-3400 Trainer. There have been tens of thousands of these sold to individuals, companies and schools. The trainer is actually part of a complete course on microcomputer fundamentals which has been very successful. (It is an excellent course; now that it is available in less expensive classroom-edition manuals, I highly recommend it to anyone who wants some more familiarity with his

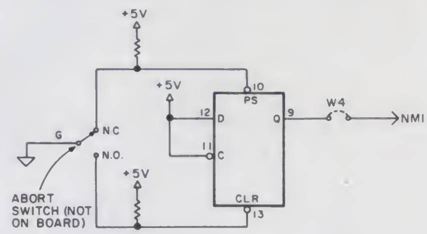


Fig. 4. Abort switch debouncing circuit on the interface board.

6800 system.)

The basic trainer has a 6800, a 1K monitor in a 6830 ROM, 256 bytes of RAM (with sockets for 256 bytes more), a 17-key numeric keyboard, six-digit seven-segment LED readout, and a prototyping socket for doing the hardware experiments in the Heath microcomputer course. The circuitry of the trainer is specially buffered to simplify hardware interfacing, and the unit runs at a clock frequency of about 400 kHz to eliminate timing problems which might otherwise hamper some of the experiments. (Heath is also marketing a less expensive trainer which lacks the prototyping socket and buffering, and is intended only for software experiments.)

Heath also sells an add-on unit to the trainer, which adds from 1K to 4K more RAM, Tiny BASIC in ROM and a serial and cassette interface. Though this makes it possible to run BASIC on the system, it does not really allow much expansion beyond that.

Several readers have written for help in expanding the ET-3400 trainer into a full-fledged SS-50 bus system and asked me to devote an article to the subject. I cannot really recommend such a step. Even if it were completed, the trainer would only act as a CPU board; external memory and I/O interfacing, as well as a motherboard and power supply, would still be needed. With the large number of 6800 CPU boards available on the used market, it would certainly be easier, and cheaper as well, to sell the Heath trainer and start with a used but real SS-50 bus system.

Conversion of the trainer to run a full-fledged system involves four aspects:

1. The 6830 ROM must be replaced by a monitor which supports a serial terminal and is compatible with existing 6800 software. Several monitors (such as MIKBUG or SWTBUG) were available in a 6830 ROM, or else a 2708 or 2716 monitor could be adapted. With a 6830 some readdressing is needed, since the trainer's 6830 EPROM occupies addresses FC00–FFFF, whereas standard monitors require E000–FFFF. With a 2708 or 2716, though, additional decoding would be needed, since the 6830 has five chip selects, while the 2708 or 2716 has only one. Either way, this would make the keyboard and LED readout on the trainer inoperable. While a custom monitor to allow both a serial terminal and the Heath I/O gear could be written, it is not worthwhile.

2. The 256-byte RAM on the board must be disabled so that a larger external memory can be substituted.

3. More extensive buffering for an external bus must be added, and some mechanical arrangement provided to mount the trainer and a motherboard together. The mechanical work would be a kludge.

4. External memory and a serial port must be added. While these could be added on the prototyping socket, there isn't enough room for more than a few K of memory; besides, a more permanent arrangement would be desirable.

6800 Systems

Although SWTP has switched production from 6800 systems to 6809 systems, there are still several manufacturers who market 6800 systems and components. Not only are 6800 systems still popular with industrial users, who need systems to go with software they already have, but with hobbyists as well.

If you are in either category, here are some companies to contact:

Febe group (51 Hamilton Ave., York, PA 17404) makes a cabinet with motherboard, power supply and keyboard for either 6800 or 6809 systems.

Gimix, Inc. (1337 West 37th Place, Chicago, IL 60609), makes a hefty cabinet, motherboard and power supply, as well as both 6800 and 6809 CPU boards and other accessories.

Thomas Instrumentation (168 Eighth St., Avalon, NJ 08202) makes motherboards and other components, including a 6802 CPU board. The 6802 is compatible with the 6800 and runs the same programs.

Percom Data Co. (211 N. Kirby, Garland, TX 75042) also makes a 6802 CPU board.

And don't forget the 6802 single board computer featured in the Kilobaud Klassroom series published in this magazine.

6800/6809 Magazines

Several times I have mentioned the *68 Micro Journal* (3018 Hamill Road, Hixson, TN 37343) in these pages. It specializes in 6800/6809 material. A newer magazine is the *SS-50 Newsletter* (PO Box 402, Logan, UT 84321). It is a slicker, easier to read magazine, whose content is improving as time goes on.■

Kilobaud Microcomputing will feature a special word processing issue in May. Manufacturers of word processing software for any microcomputer who want to be included in this issue should contact the *Microcomputing* editoral department immediately. Please send spec sheets of your software system. We will contact you if we need additional information.

✓ Reader Service—see page 178

Protect Your Files From Prying Eyes

By Phil Hughes

Most microcomputers offer little protection against unauthorized access to data. There is, however, a way to ensure data privacy. Encryption will change a message into a form that is unintelligible to anyone without the converting information.

Most of us probably played with a simple form of encryption when we were children. For example, we might have written a note by converting each letter of the alphabet to the letter following it. Thus,

Mary, I like you a lot. John

became

NBSZ, J MJLF ZPV B MPU. KPIO.

Although this message is obscure at first, it is easy to decipher. The single letters J and B are the clues. English has only two single-letter words, so B and J represent A and I. Decoding should take only a couple of minutes.

Generating Encrypted Messages

With a computer system, you should be able to generate encrypted messages that are much more difficult to decrypt.

One simple method is to use an exclusive OR (XOR) to replace each character in the message with a character in a key (see Table 1). If this is done with a message in ASCII code, an assortment of printable and nonprintable (control) characters is produced.

For example, encrypt the word MARY using the key 3ZRB. First convert the ASCII characters to their binary equivalents. Using a conversion table, you get Example 1.

Phil Hughes, PO Box 2847, Olympia, WA 98507.

If you execute an exclusive OR on these two lines, the result is shown in Example 2.

Now comes the real advantage of XOR. The XOR operation is symmetrical; applying it a second time will undo what it did the first time. See Example 3.

If you try to apply this encryption method, you may run into trouble with what the operating system does with the control codes that are generated. This happened to me with the Flex operating system. I decided, therefore, to only encrypt the printable characters, and leave the control codes, such as carriage return, as they were.

Thus, I also had to cause the resulting characters to have values greater than the valid control codes. My final algorithm is shown in Example 4.

Also, the length of the key is generally shorter than the message. I handled this by reusing the characters in the key as many times as required to complete the encryption. For example, if the message is

THIS IS A SECRET

and the key is

HIDE

then the character-by-character XOR is performed by repeating HIDE the required number of times. Thus,

THIS IS A SECRET!
HIDEHIDEHIDEHIDEH

Programs

Listing 1 is an implementation of this encryption method in 6800 assembler language. Most of the program is for interfacing to the file management system, getting the parameters and checking for errors. The actual encryption takes place in lines 209–220.

Listing 2 is an example using Crypt. First, using the build utility, you create a data file named Message. You then list the contents of the file, using

```
MARY  =  0100 1101 0100 0001 0101 0010  0101 1001
3ZRB  =  0011 0011 0101 1010 0101 0010 0100 0010
```

Example 1. Converting ASCII characters to their binary equivalents.

```
0111 1110 0001 1011 0000 0000 0001 1011
```

Example 2. Executing an exclusive OR.

```
result  =  0111 1110 0001 1011 0000 0000 0001 1011
3ZRB    =  0011 0011 0101 1010 0101 0010 0100 0010

result in:
           0100 1101 0100 0001 0101 0010 0101 1001

which is MARY.
```

Example 3. Applying XOR to the result in Example 2, using the encryption key.

```
IF NOT control character THEN
Subtract hexadecimal 20 from character
Perform XOR
Add hexadecimal 20 to result
```

Example 4.

```
+++BUILD MESSAGE
  =This is the message.
  =It will be encrypted and then it will
  =be decrypted.
  =Watch what happens.
  =#
+++LIST MESSAGE
This is the message.
It will be encrypted and then it will
be decrypted.
Watch what happens.

+++CRYPT,MESSAGE,HIDDEN,FRAMUS
+++LIST HIDDEN
:(>u:5r5%0s+72>44#!
!s1;-!u1#r$#6!?'5(1s'<%m!;#<a$!s1;-!
1#r%(6!?'5(1}
 96;f%),!s.31=0=5!

+++CRYPT,HIDDEN,FOUND,FRAMUS
+++LIST FOUND
This is the message.
It will be encrypted and then it will
be decrypted.
Watch what happens.
```

Listing 2. Using Crypt.

```
+++CRYPT,MESSAGE,F1,WEIRD
+++CRYPT,F1,F2,STRANGE
+++LIST F2
Pyr`*ys:r~s+crdpvoy-
_i rpn`!r`$b{bvhkgot {hr6fry#~!<t{za
g!'hdsw}wad`?
Dkdcr&a~Jz7bgxyma8


+++CRYPT,F2,G1,STRANGE
+++CRYPT,G1,G2,WEIRD
+++LIST G2
This is the message.
It will be encrypted and then it will
be decrypted.
Watch what happens.


+++CRYPT,F2,H1,WEIRD
+++CRYPT,H1,H2,STRANGE
+++LIST H2
This is the message.
It will be encrypted and then it will
be decrypted.
Watch what happens.
```

Listing 3. Encrypting Message, using the key Weird and the key Strange.

the list utility. Next, you encrypt Message into the file Hidden, using Framus as the key. The contents of Hidden show little resemblance to the original contents of Message.

Finally, Crypt is run once again, this time using Hidden as input, Found as output and—again—Framus as the key. Listing Found shows that the message has been restored to its original form.

For those wishing to make the message even harder to decode, Crypt can be repeated using different keys. Listing 3 shows Message being encrypted, first using the key Weird and then the key Strange. Because of the symmetry of the XOR operation, the decryption can be performed in either order. This is also demonstrated in Listing 3.■

Listing 1. Implementation of encryption method in 6800 assembler language.

```
        0000                          NAM    CRYPT     - File encryption utili
                                                         ty
        0000                          OPT    XRF
                           * Copyright 1979, Specialized Systems Consultants
                           *
                           * COMMAND:
                           *   CRYPT,filespec1,filespec2,key
                           *
                           *   where filespec1 is the input file
                           *         filespec2 is the output file and
                           *         key is the encryption key
                           *
        0000                          OPT    PNT
                           *
        0078               KBLEN      EQU    120       key buffer length
                           *
                           *
        0000               CRYPT      EQU    *
 3R     0000 CE 0154                  LDX    #RFCB
12 *    0003 BD AD2D                  JSR    GETFIL    input file
16      0006 24 03 (000B)             BCC    $10
 3R     0008 7E 00A6                  JMP    ERROR
 2R     000B 86 01         $10        LDA A  #FEXT.TXT default e,t
11 *    000D BD AD33                  JSR    SETEXT
14      0010 CE 0294                  LDX    #WFCB
23 *    0013 BD AD2D                  JSR    GETFIL    output file
27      0016 24 03 (001B)             BCC    $99
 3R     0018 7E 00A6                  JMP    ERROR
 2R     001B 86 01         $99        LDA A  #FEXT.TXT default ext
11 *    001D BD AD33                  JSR    SETEXT
                           * Get the key
14      0020 CE 00DC                  LDX    #KEYBUF
16      0023 5F                       CLR B
25 *    0024 BD AD27                  JSR    NXTCH
29      0027 24 0E (0037)             BCC    GKOK
 3R     0029 CE 00CA       BADKEY     LDX    #BKMSG    bad key
12 *    002C BD AD1E                  JSR    PSTRNG
15      002F 7E AD03                  JMP    WARMS
 9R*    0032 BD AD27       GKLOOP     JSR    NXTCH     get character
13      0035 25 0A (0041)             BCS    GKDONE    end of key
 2R     0037 5C            GKOK       INC B
 4      0038 C1 78                    CMP B  #KBLEN    too long?
 8      003A 2C ED (0029)             BGE    BADKEY    yes
 6R     003C A7 00                    STA A  0,X       save char
10      003E 08                       INX
14      003F 20 F1 (0032)             BRA    GKLOOP    get next char
 2R     0041 81 0D         GKDONE     CMP A  #X'D'     valid terminator?
 6      0043 27 05 (004A)             BEQ    GKTERM    yes
 4R     0045 B1 AC02                  CMP A  F2EOL
 8      0048 26 DF (0029)             BNE    BADKEY    no
 5R     004A F7 00D9       GKTERM     STA B  KEYLEN    save
                           * Open the files
 8      004D CE 0154                  LDX    #RFCB
```

More

XOR	0	1
0	0	1
1	1	0

Table 1. Truth table for exclusive OR (XOR). The result is 1 if the two inputs have different values and 0 if the two inputs are the same.

Listing 1 continued.

```
190  10    0050 86 01                        LDA A #FM2OPR     read
191  16    0052 A7 00                        STA A FB2FNC,X
192  25 *  0054 BD B406                      JSR   FMS
193  29    0057 26 4D (00A6)                 BNE   ERROR
194   2R   0059 86 FF                        LDA A #X'FF'      disable
195   8    005B A7 3B                        STA A FB2SCF,X    space comp
196  11    005D CE 0294                      LDX   #WFCB
197  13    0060 86 02                        LDA A #FM2OPW     write
198  19    0062 A7 00                        STA A FB2FNC,X
199  28 *  0064 BD B406                      JSR   FMS

200  32    0067 26 3D (00A6)                 BNE   ERROR
201   2R   0069 86 FF                        LDA A #X'FF'      disable
202   8    006B A7 3B                        STA A FB2SCF,X    space comp
203  11    006D CE 00DC                      LDX   #KEYBUF     point to key
204  17    0070 FF 00DA                      STX   KBPTR
205  21    0073 F6 00D9                      LDA B KEYLEN      key char count
206   3R   0076 CE 0154         LOOP         LDX   #RFCB
207  12 *  0079 BD B406                      JSR   FMS         get char
208  16    007C 26 22 (00A0)                 BNE   CHKSTAT
209   5R   007E FE 00DA                      LDX   KBPTR
210   7    0081 81 20                        CMP A #X'20'
211  11    0083 2B 06 (008B)                 BMI   $40
212   2R   0085 80 20                        SUB A #X'20'
213   7    0087 A8 00                        EOR A 0,X
214   9    0089 8B 20                        ADD A #X'20'
215   4R   008B 08              $40          INX
216   6    008C 5A                           DEC B
217  10    008D 26 06 (0095)                 BNE   SVPTR
218   4R   008F F6 00D9                      LDA B KEYLEN      restore key length
219   7    0092 CE 00DC                      LDX   #KEYBUF     and buffer ptr

220   6R   0095 FF 00DA         SVPTR        STX   KBPTR
221   9    0098 CE 0294                      LDX   #WFCB
222  18 *  009B BD B406                      JSR   FMS         put char
223  22    009E 20 D6 (0076)                 BRA   LOOP        get next
224                             * Check status
225   5R   00A0 E6 01           CHKSTAT      LDA B FB2ESB,X    get status
226   7    00A2 C1 08                        CMP B #8          EOF?
227  11    00A4 27 03 (00A9)                 BEQ   DONE
228   9R*  00A6 BD AD3F         ERROR        JSR   RPTERR      report error
229                             * Close files

230   3R   00A9 CE 0154         DONE         LDX   #RFCB
231   5    00AC 86 04                        LDA A #FM2CLF
232  11    00AE A7 00                        STA A FB2FNC,X
233  20 *  00B0 BD B406                      JSR   FMS
234  24    00B3 27 03 (00B8)                 BEQ   $20
235   9R*  00B5 BD AD3F                      JSR   RPTERR      close error
236   3R   00B8 CE 0294         $20          LDX   #WFCB
237   5    00BB 86 04                        LDA A #FM2CLF
238  11    00BD A7 00                        STA A FB2FNC,X
239  20 *  00BF BD B406                      JSR   FMS
240  24    00C2 27 03 (00C7)                 BEQ   $40
241   9R*  00C4 BD AD3F                      JSR   RPTERR      close error
242   3R   00C7 7E AD03         $40          JMP   WARMS       return to O.S.
243                             *
244                             * DATA AREA
245                             *
246        00CA 2A2A20494E56    BKMSG        CON   C'** INVALID KEY',4
247        00D9                 KEYLEN       RMB   1           key length
248        00DA                 KBPTR        RMB   2           buffer pointer
249        00DC                 KEYBUF       RMB   KBLEN       key buffer
250        0154                 RFCB         RMB   320         input FCB
251        0294                 WFCB         RMB   320         output FCB
252                             *
253        03D4                              END   CRYPT

AD36 ADDBX       137
0029 BADKEY      172    179   186
00CA BKMSG       246    172

00A0 CHKSTAT     225    208
AD21 CLASS       130
AD00 COLDS       119
0000 CRYPT       154    253
AD4B DOCMND      144
00A9 DONE        230    227
00A6 ERROR       228    158   164   193   200

AC00 F2BACK       78
AC29 F2CCOL      109
AC28 F2COMF      108
AC18 F2CURC       96
AC07 F2DEC        85
AC01 F2DEL        79
AC03 F2DEPC       81
AC08 F2EJC        86
AC2D F2ENV       111
AC02 F2EOL        80    185
AC16 F2ERR        95
AC20 F2ERRT      102
AC0A F2ESC        88

A840 F2FCB       113

AC26 F2FIA       107
AC2F F2FIEF      112
AC24 F2FOA       106
AC23 F2INSW      105
AC11 F2LAST       92
AC14 F2LBP        94
A080 F2LBUF       77
AC1B F2LDAO       99
AC1A F2LINE       98
AC2B F2MEND      110
AC05 F2NULL       83
AC22 F2OTSW      104
AC09 F2PAUS       87
AC19 F2PREC       97
```

DR. DALEY OFFERS SOFTWARE FOR EVERYONE

DATA BASE

The data base package allows total user control over the contents of each entry in the file. Features user selectable record size from 5 to 242 characters per record, statistical and plotting package, output with WORDPRO files or printer. Includes full user definable output formatting. With optional indexing routine can produce a comprehensive index of a data set.

$299.95
Index 99.95

MAIL LIST

This powerful mailing list package features a variety of options for producing labels. It includes user defined file structure and label format. Label format can list to the printer or to WORDPRO format files.

$159.95

For PET or CBM 2000 or 8000 series with 32K memory please specify your machine configuration.

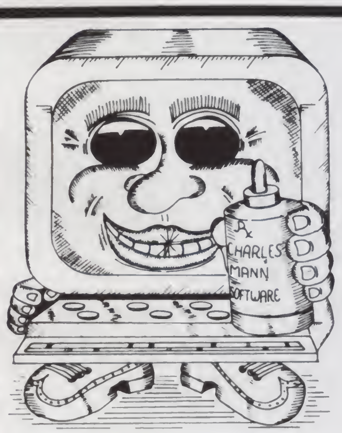
SOFTWARE LIBRARY

Hundreds of schools and individuals have purchased this package for use as an educational tool or just plain fun. It contains 50 (yes fifty!) programs. This ranges from our famous TREK 3 and horse race to fun learning programs for children to checkbook and a micro mail list program with lots in between. At about $1.40 per program how can you miss?

Cassette $69.95
Diskette 79.95
For APPLE II or PET

Charge to your MC/VISA

DR. DALEY'S SOFTWARE ✓34
425 Grove, Berrien Springs, Michigan 49103
Phone (616) 471-5514 Sunday-Thursday noon to 9 p.m. Eastern Time



CURE TO SOFTWARE PROBLEMS

PROFESSIONAL SOFTWARE

Medical, Dental & Legal Systems, Accounting & Financial, Educational, Word Processing, Office Management

Check your Local Dealer or Contact:

Charles Mann & Associates ✓165
7594 San Remo Trail
Yucca Valley, Ca. 92284
(714) 365-9718

Apple II TRS-80 TI 99/4

MEMOREX
DISKETTES & CARTRIDGES

for your computer or word processor

BUY THE BEST FOR LESS. Lowest prices. **WE WILL NOT BE UNDERSOLD!!** Buy any quantity. Call free (800) 235-4137 for prices and information.

PACIFIC EXCHANGES ✓246
100 Foothill Blvd.
San Luis Obispo, CA
93401. (In Cal. call
(805) 543-1037.)

Listing 1 continued.

```
AC0E F2SDR      91
AC0B F2SDRV     89
AC21 F2SIOF    103
AC06 F2TABC     84
AC1D F2TRAN    100
AC1E F2TRNA    101
A100 F2UCA     114
AC12 F2UCTA     93
AC0C F2WDRV     90
AC04 F2WIDC     82
0002 FB2ACT     41
0040 FB2BUF     59
002F FB2CDA     56
0020 FB2CRN     52
001E FB2CUR     51
0003 FB2DRV     42
0013 FB2EDA     47
0001 FB2ESB     40   225
000C FB2EXT     44
000F FB2FAT     45
0032 FB2FDP     57
001C FB2FLP     50
0000 FB2FNC     39   191  198  232  238
0022 FB2INX     53
0004 FB2NAM     43
0024 FB2NWB     55
0023 FB2RDX     54
003B FB2SCF     58   195  202
0011 FB2SDA     46
0015 FB2SIZ     48
0017 FB2SMI     49
0008 FEXT.BAC   70
0005 FEXT.BAK   67
0003 FEXT.BAS   65
0000 FEXT.BIN   62
0002 FEXT.CMD   64
0007 FEXT.DAT   69
0009 FEXT.DIR   71
000B FEXT.OUT   73
000A FEXT.PRT   72
0006 FEXT.SCR   68
0004 FEXT.SYS   66
0001 FEXT.TXT   63   159  165
0000 FL2IO      16
0016 FM2BOR     35
0004 FM2CLF     20   231  237
000C FM2DLF     27
0014 FM2FND     33
0007 FM2GIR     23
0011 FM2GRB     31
000F FM2NSS     29
0006 FM2OPD     22
0001 FM2OPR     17   190
0003 FM2OPU     19
0002 FM2OPW     18   197
0010 FM2OSI     30
0008 FM2PIR     24
0015 FM2POS     34
0012 FM2PRB     32
000D FM2RNF     28
0009 FM2RSS     25
0005 FM2RWF     21
000A FM2WSS     26
B400 FMINIT    145
B406 FMS       147   192  199  207  222  233  239
B403 FMSCLS    146
AD15 GETCHR    126
AD2D GETFIL    134   156  162
AD42 GETHEX    141
0041 GKDONE    183   176
0032 GKLOOP    175   182
0037 GKOK      177   171
004A GKTERM    187   184
AD1B INBUFF    128
AD09 INCH      122
AD0C INCH2     123
AD48 INDEC     143
0078 KBLEN     151   178  249
00DA KBPTR     248   204  209  220
00DC KEYBUF    249   168  203  219
00D9 KEYLEN    247   187  205  218
AD30 LOAD      135
0076 LOOP      206   223
AD27 NXTCH     132   170  175
AD45 OUTADR    142
AD0F OUTCH     124
AD12 OUTCH2    125
AD39 OUTDEC    138
AD3C OUTHEX    139
AD24 PCRLF     131
AD1E PSTRNG    129   173
AD18 PUTCHR    127
AD06 RENTER    121
0154 RFCB      250   155  189  206  230
AD3F RPTERR    140   228  235  241
AD2A RSTRIO    133
AD33 SETEXT    136   160  166
0095 SVPTR     220   217
AD03 WARMS     120   174  242
0294 WFCB      251   161  196  221  236
```

SAVE!

WITH DSC'S 10% CLUB

- **YOU CAN BUY HARDWARE, SOFTWARE, AND SUPPLIES AT 10% ABOVE WHOLESALE**
- **JOIN NOW: $15.00 ANNUAL MEMBERSHIP FEE INCLUDES $10.00 CREDIT TOWARD FIRST PURCHASE AND CATALOG OF PRODUCTS**

- TERMS — PREPAYMENT, CASH, CERTIFIED CHECK, M.O., M.C. AND VISA ADD 4%, FOB PHILA., PA.
- WARRANTIES — ALL EQUIPMENT CARRIES MANUFACTURER'S WARRANTY AND WILL BE SERVICED BY MANUFACTURER OR REPRESENTATIVE

COMPUTERS	
Intertec - SuperBrain 32K	2375.
SuperBrain 64K	2500.
SuperBrain QD 64K	2990.
Intertube III	724.
Atari - 400 Computer	475.
800 Computer	746.
815 Dual Disk Drive	1159.
Apple II + Apple II Plus - 16K	959.
32K	1049.
48K	1149.
Drives - Lobo for Apple - 5¼" with controller	442.50
VR Data - 40 tk 1 side	309.
80 tk 1 side	439.
40 tk 2 side	439.
80 tk 2 side	619.
8" CDC 9404B, Mod II comp.	749.
16K memory	28.95
Modems - D-Cat direct connect	150.
Lexicon acoustic	129.95
Radio Shack - Mod. I Expansion Interface	246.40
Model III 16K	869.
Model III 32K w/2 Drives	2199.
Color Computer 4K	345.
Color Computer 16K	520.
PRINTERS	
Centronics - 730-1 parallel	594.
730-3 serial	649.
737-1 parallel	715.
737-3 serial	799.
704-9 RS-232 serial	1599.
Anadex - 8000	794.
NEC - 5510 w/tractor	2490.
C. Itoh Starwriter 25 cps parallel	1279.
SUPPLIES	
Verbatim Diskettes - 5¼" MD-01 Data Life	26.90/10
550-01, DD	37.50/10
8" FD-34-9000-Data Life	32.40/10
Nashua Diskettes - 5¼" (high quality)	22.95/10
8" DD, FD-1D-WP	32.50/10
Dysan - 5¼" SS-SD	24.50/5

ALL BRANDS OF HARDWARE, SOFTWARE, PERIPHERALS, SUPPLIES AND BOOKS ARE AVAILABLE.

I WANT TO JOIN NOW.

NAME ______________________
STREET ______________________
CITY ____________ STATE ________ ZIP ________
CHARGE ☐ MASTER CHG. ☐ VISA
CARD NUMBER ______________________
EXPIRATION DATE ______________________

DISK SUPPLY CO.

SUITE 439, 111 S. OLIVE ST. MEDIA, PA 19063
(215) 461-5437

PACIFIC OFFICE SYSTEMS

COMPUTER INTERFACES & PERIPHERALS

- **POS-100 NRZ1 TAPE DRIVE CONTROLLER/FORMATTER** — Now your micro can read and write IBM/ANSI compatible NRZ1 format 9-track magnetic tapes. The POS-100 consists of S-100 bus card, 6' ribbon cable, tape drive controller card, cable to Pertec-Standard NRZ1 Tape drive, plus documentation and Z-80 or 8080 software (specifiy). Power is derived from tape drive and S-100 bus. Ship Wt.: 10 lbs. Suggested Retail Price $995.00
- **POS 103/202 "MIX or MATCH" MODEM** — Unique POS control design permits use in one housing of both Bell-compatible 103 (0-300 baud) and 202 (0-1200 baud) modem modules originally made by VADIC Corp. for a telephone company subsidiary. FEATURES: RS-232 serial Interface, auto-answer, auto-dial, LED display, telephone line interface via acoustic coupler, manual DAA, or auto-answer DAA (sold separately). FULLY ADJUSTED; no special tools required. 3,000 mile range over standard dial-up telephone lines. Ship wt.: 15 lbs.
  PRICES . . . . . POS 103 Modem — $199.95; POS 202 Modem — $299.95; POS 202 Modem w/Auto-Answer — $349.95; POS 103/202 Modem — $499.95; FCC-Approved Auto-Answer DAA — $125.00; Acoustic Coupler — $29.95.
- **POS DAISY-WHEEL PRINTER INTERFACE for TRS-80** — Will drive Diablo HyType I, HyType II, and Qume Q and Sprint 3 printers. Includes 1K user-available memory for custom print routines (such as graphics, bidirectional printing, etc.). Programmed to respond to print commands from BASIC ELECTRIC PENCIL™, and SCRIPSIT™ software. Draws its power from printer. Ship wt.: 5 lbs. Price $250.00
  Cables, each (Specify HyType I, HyType II, or Qume) . . . . . $ 25.00
- **POS ASCII INTERFACE for IBM I/O SELECTRIC** — This Centronics-style parallel printer interface will drive an IBM Model 731 or 735 I/O typewriter (EBCD and Correspondence codes). No software needed. Features on-board EPROM which holds up to 8 ASCII-to-IBM code tables for different type spheres. Closed-loop operation runs at maximum printer speed; stops and starts on a single character without loss of data. Requires +12VDC and ±5VDC power source. Ship wt.: 5 lbs. Price $249.95
  Power Supply (±5VDC, +12VDC, +24VDC for Solenoids on Printer) . . . . $ 49.95
- **CONVERT OFFICE SELECTRIC TO I/O TYPEWRITER** — Kit includes assembled solenoids, switches, wire harness, magnet driver PCB plus instructions for installation and mCPU interface. Price $150.00
- **"FORMALINER"** Variable Width Forms Tractor for 15" Selectrics . . . $95.00
- **GTE Model 560 ASCII SELECTRIC I/O Terminal** — With RS-232 Serial Interface and digital cassette deck for use as memory typewriter. Ship wt.: 100 lbs. Price, tested and adjusted $1,195.00
- **POS ASCII IBM SELECTRIC PRINTER** — 15" Selectric from GTE terminal cleaned and adjusted with POS Centronics-style ASCII printer interface. UC/LC, carbon and fabric ribbons. Compatible with TRS-80, Apple, SOL and other CPU parallel printer ports. Ship wt.: 75 lbs. Price $895.00

PACIFIC OFFICE SYSTEMS ✓153
918 Industrial Avenue Palo Alto, CA 94303 (415) 493-7455

FOR OSI COMPUTERS

★ SMARTERM ★

An 'Intelligent' Terminal Software System for OSI Computers by Phil Lindquist

- Even, Odd, or No Parity Use-Selected
- Full or Half Duplex Operation
- User Controlled Storage of Incoming Data
- Stored Data Printed or Saved on Disk
- Saved Data Recalled For Review or Output
- Control Key Initiation of LOGON Messages

- Join The Micro Communications Revolution -

C2-4P MF, C4P MF Systems $19.95
C2-8P DF, C8P DF, C4P DF Systems $23.95

Order From:

COMPUTER CONNECTION, INC.
38437 Grand River • Farmington Hills, Mich. 48018
✓278

Dealer Inquiries Welcome

✓ *Reader Service—see page 178*

A simple hardware modification lets you save and load at 600 baud.

Faster Baud Rate For the Superboard II Cassette

By James A. Antonelli

Being the owner of a Superboard II computer has its drawbacks. The hardware and software people seem never to have heard of OSI, and magazine articles are few. So when I wanted to speed up my program-loading time, I found that I had to do it myself.

Some of my programs are 8K long. At 300 baud, loading can take as long as five minutes. I could have gone to disk but didn't have the money. So after studying the schematics in the user's manual, I came up with this idea.

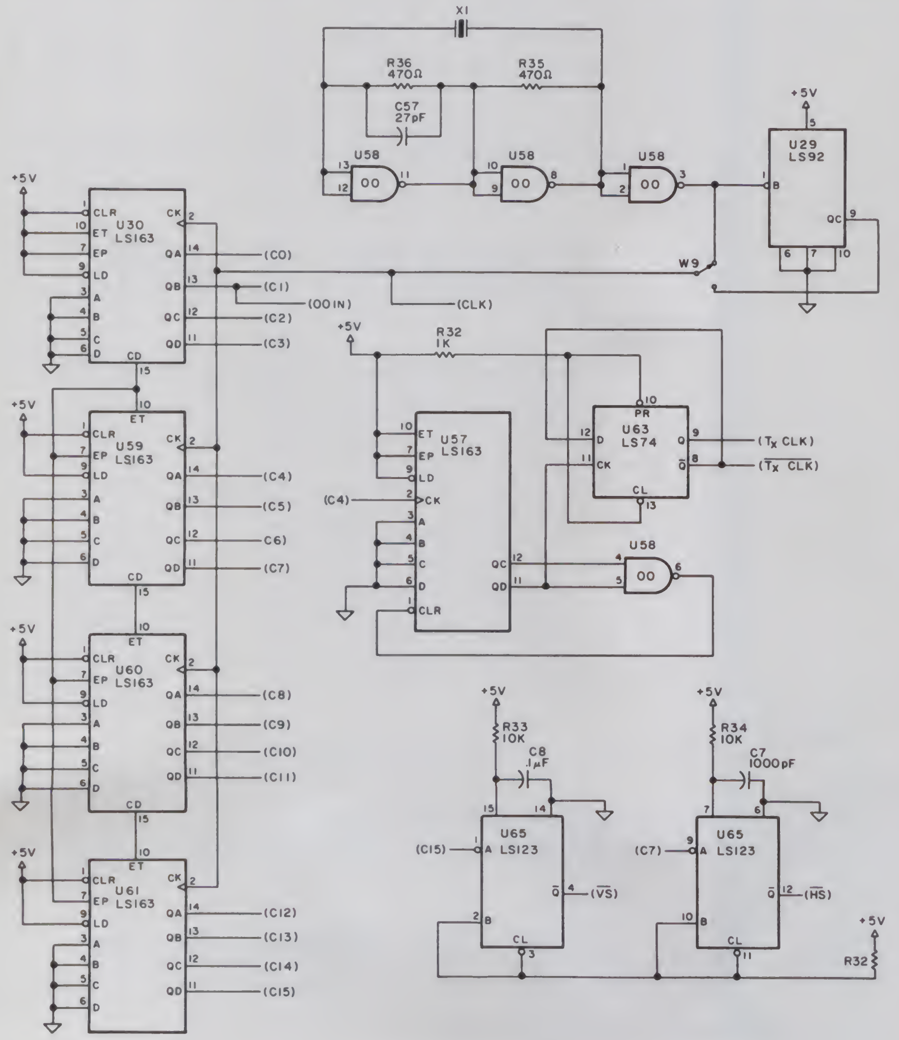


Fig. 1.

The Circuit

As can be seen in Fig. 1, the clock circuit U58 is divided into separate clock frequencies by U30, U59, U60 and U61. The chips we will be concerned with are U30, U57 and U59. Chip U57 receives clocking from U59, as can be seen in Fig. 1. Pin 2 of U57 connects to pin 14 of U59; this setup gives us the 300 baud rate.

First, to double the frequency or baud rate, cut the foil running to pin 2 near U57. Exercise care when cutting, to avoid damaging other traces.

When this foil is cut and all material is removed from the 600 board, you are then ready to proceed to the next step. You will need shielded cable, wire cutters and a small soldering iron (see parts list). If you also want to keep the 300 baud rate, then skip step 2 and go on to step 3.

Second, for a 600 baud rate only, you need to install a jumper wire from U30, pin 11, to U57, pin 2. You will also have to remove R53 and install a jumper here. It is advisable to ground the shielded cable at the nearest ground. Pins 3, 4, 5 and 6 of U57 are all grounded so you can ground the cable here. You can ground each end of the cable if you wish. I only grounded one end and haven't run into any problems yet.

After you have completed steps 1 and 2, skip to the alignment section,

Address correspondence to James A. Antonelli, Box 263 RD 1, Sugarloaf, PA 18249.

step 4.

Step 3 requires a little more work, but you will have a choice between 300 and 600 baud. You will have to determine where you want to mount SW1. I mounted mine in the front, above the keyboard. Then measure the single conductor shielded cable from here to U57, leaving a few extra inches. Cut and strip insulation and tin ends. Solder this to the center position of SW1.

Now measure your second wire. Cut and strip insulation and tin ends. This one goes to U30. You can now solder this wire to the switch also, either to the top or to the bottom (see Fig. 2).

Now measure your last wire. This one goes to U59. Cut and strip insulation and tin ends. It is a good idea to tape all of the wires together. It makes a neater project, and servicing is easier.

One other note: make sure you identify the wires. Either mark them with a colored tape, wire numbers or any other method that will make them easy to identify. Solder this wire in the remaining lug on the switch, which should now only have three lugs left. These should all be on the opposite side. If not, go back to the beginning of step 3. It is a good idea to now go over your solder joints.

Cut two pieces of solid wire about six inches long. Strip about one inch of insulation. Wrap one of these wires around SW1, the bottom lug. You also should have one of the wires from the two-conductor cables on this lug. Solder these in place. The opposite end of this wire goes to the end lug of R1. With the wire remaining, twist this and the other wire from the two-conductor cable to the center lug of SW1 and the end lug of R1. This completes the switch.

The next step is to locate and remove R53 on the board (see Fig. 3). It is located near U69, pin 7.

The only wire left is the two-conductor shielded cable. Solder one of these wires, after cleaning the holes from the resistor R53. Don't forget to solder the top and bottom of the board. These are plated-through holes, but sometimes the plate is damaged when removing parts. The wire remaining can now be soldered in place.

This completes the board. Check for solder bridges and/or bad solder joints. If everything is OK, then lay the switch on the side and cut one more piece of solid wire to about six inches in length. Strip all but three inches of insulation.

On the edge of the 600 board there should be a ground, and a few holes through this ground strip. If your particular board doesn't have these holes, you'll need to drill two holes here. Now insert the solid wire from the top, and after bending the wire bring this wire back up through the remaining hole. This will be used for holding the wires that were soldered previously. Route all wires to this point and, with the remaining solid wire, wrap this. This will keep the wires from being pulled off the foil. Your finished product should look quite professional.

The Alignment

Fourth, set the switch on nonconductive material. Turn your computer on; you should see the LED light up. If not, shut it off and recheck. If you have a frequency counter, alignment will be easy. If not, then enter the following simple program:

```
10 For I=0 to 32: For A=0 to 24: print "A";: next A: next I.
```

Now type in the command word SAVE, and run. Depending on the position of SW1, it will print letters A at the regular speed of 300 baud, or they will be printed at 600 baud.

As this program is being run, switch SW1 back and forth. There should be a difference in speed at which the As are being printed. Once you know which position is 300 baud (it should be in the up position), mark it. It doesn't really make any difference where the 300 or the 600 baud is, just so you will know.

Get a blank tape and make a change in the program to:

```
10 For I=0 to 100: For A=0 to 24: print "A";: NEXT A: NEXT I
```

This should give enough recording time. Set SW1 to 600 baud, start the tape recorder and let the program run until it is completed. Flip SW1 to 300 baud and start the tape recorder. Let the program run until it is also completed.

You should now have two recordings, one at 600 baud and one at 300 baud. Rewind the tape and type in NEW, and enter this program:

```
10 Load: INPUT A$: GOTO 10.
```

One resistor, R57, has to be located on the 600 board. On my particular board, it is the blue one located near the rear of the board. Once you have located this resistor, enter RUN on the computer and start the tape recorder. *Slowly* turn until you start getting As printed on your monitor.

Once you get a consistent load (none of the As are being dropped out and you haven't received a syntax error message), leave R57 alone. Repeat the above load program a few times to be sure of the 600 baud rate. When you believe everything is OK, then let the tape advance to the second recording and set SW1 to 300 baud rate. Once again type in RUN. You will now have to adjust R1 until

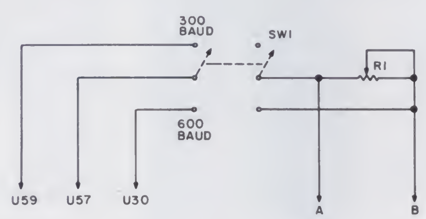


Fig. 2.

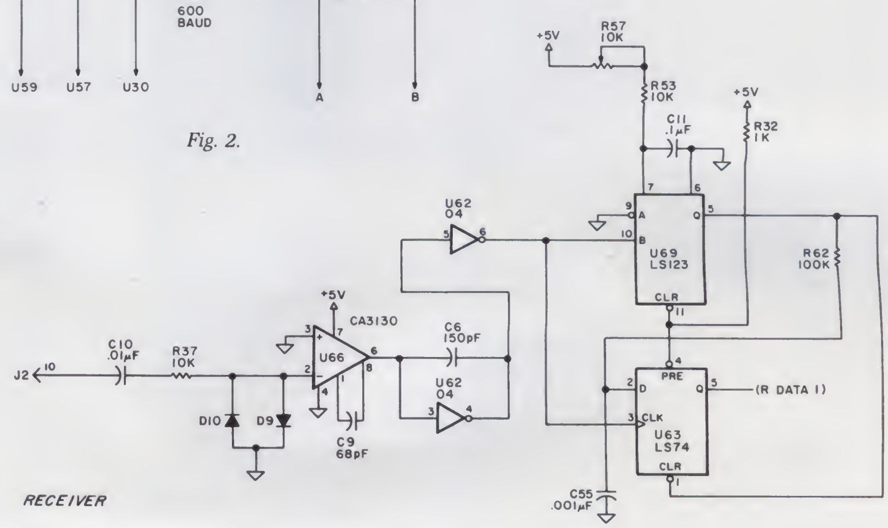


Fig. 3

you get a consistent loading of As. Redo until you think everything is OK. Flip SW1 to 600 baud and try this position again. Sometimes you'll need to make very small adjustments to R57, but remember: any adjustment to this resistor also means an adjustment to R1. If everything is OK, then mount SW1 and enjoy rerecording your programs at double the speed.

I made this conversion nearly a year ago, and I haven't had to make any adjustments whatsoever. I highly recommend using a quality tape for the 600 baud rate. Using cheap tape may result in drop-outs; you then have to reload with your back-up tape (300 baud) and have gained nothing! ■

Parts List

- Two-conductor shielded cable
- Single-conductor shielded cable
- 10k mini trimpot
- 2PDT mini switch
- Stranded hook up wire
- Solid hook up wire
- Small soldering iron, solder, etc.

Any microcomputerist can be a business opportunist with this mailing label scheme.

An Adventure In Free Enterprise

By Forest E. Myers

It started innocently enough. I had just finished a general program that could, among other things, generate mailing labels from a DATBAS file. My brother-in-law asked if I would be willing to do some labels for a friend. A week or two later, the friend called asking about prices. Needless to say, I was taken aback by the prospect of having a paying customer. Thus began my adventure in the mailing label business.

The program, which serves as the focus of this article, is called Labels, and is an extension of two previously-published programs: Screen (*Microcomputing*, April 1980) and Fileit (*Microcomputing*, May 1980). It lets you print more than one DATBAS record across a page. This reduces the amount of paper used when dumping a data file. By making the program more general, selected items can be output from data base records. Presto, change-o, a mailing label generator.

A Market for Labels

If you've looked closely at your mail lately, you have seen the extent to which gummed labels are used. There are business firms that specialize in mailing list maintenance, update and envelope services. For large-volume users, these companies offer a very cost-effective method of relieving the burden associated with large mailings.

For small-volume users, however, these services are prohibitively expensive. The large services have a required minimum on the number of labels in a run. As a result, the small mailer must buy a several-months' supply of address labels to reach the minimum level. Invariably, addressees change over time and the labels become stale.

Therefore, the small mailer must hand-write address labels or use copy machine labels. The former takes an inordinate amount of time. The latter is difficult to update and hard to maintain in any prearranged sequence.

A niche, therefore, exists for the microcomputer owner. There is a market to be supplied, one that consists of real estate agents, manufacturers' representatives, churches, bowling leagues, small businesses and so on.

To tap this market requires a little hustling. Contact friends and tell them about your service. Visit local establishments and, if the budget will permit, take out occasional newspaper ads. Look for trade magazines whose readers could be potential customers. Many times the advertisement rates in these publications are low and give you wide exposure for little cost. Mailings are another tool for telling others of your entry into the mailing label business.

Address correspondence to Forest E. Myers, 5114 Garnett St., Shawnee, KS 66203.

Program listing.

```
10  FILL 12890,145 : REM LET BASIC PRINT OUT A 145 CHARACTER LINE
20  CLOSE (0,E) : CLOSE (1,E) : CLOSE (2,E) : CLOSE (3,E)
30  DIM A$(256),B$(10),C$(10),D$(256),E$(512),C(30),F(30)
40  DIM R(30),L(30),F$(256)
50  DIM G$(256),T$(256)
60  DIM R$(256),L1(30)
70  DIM D(30),N1(30)
80  DIM D1(30),T(30)
90  #"" : REM CLEAR SCREEN AND HOME CURSOR
100  FOR I=1 TO 7 : #" " : NEXT I
110  #"                         LABEL PROGRAM"
120  A$=" " : FOR I=1 TO 8 : A$=A$+A$ : NEXT I
130  D$=A$ : B$=A$(1,10) : C$=B$ : E$=A$+D$ : F$=A$
140  G$=A$ : T$=A$
150  R$=A$
160  R$=A$(1,1)
170  FOR I=1 TO 5 : #" " : NEXT I
180  INPUT "Enter DATBAS file name ",B$
190  INPUT "Enter drive no. where DATBAS file can be found ",D1
200  B$=B$+".DA"+CHR$(0)
210  OPEN (2,E,B$,2,D1)
220  GET (2,E,A$,0)
230  C$=A$(241,242) : CONVERT C$ TO Z : REM NUMBER OF LABELS IN FILE
240  C$=A$(243,244) : CONVERT C$ TO L1 : REM STARTING CRT LINE
250  C$=A$(245,246) : CONVERT C$ TO B1 : REM BLOCKING FACTOR
260  C$=A$(247,249) : CONVERT C$ TO D : REM LENGTH OF RECORD
270  C$=A$(250,252) : CONVERT C$ TO Z3 : REM NUMBER OF RECORDS IN FILE
```

More

However, before spending a lot of time, effort and money in this area, consider the likelihood that your envelope will be thrown in the trash before it is opened. My own experience tells me never to send letters to real estate agents. They are bombarded with every type of sales pitch. Letters stand little chance of spurring interest. A visit to the office may produce more results. Be careful, though. Your visits may be viewed as business soliciting, and you may run afoul of local laws prohibiting soliciting without a license. Check with city officials before you make your calls.

Finally, before you visit, work up a small sales pitch and practice it on an unsuspecting friend, wife or family pet so you don't waste your potential customers' time. Also, work up a small reminder card or sales poster so if they want to use your service in the future, they will know where to reach you.

Pricing the Service

Having tackled the problem of contacting potential customers, how do you price your service? Depending upon the extent of your services, you can levy charges in several areas. The first is a data entry charge, which is for entering the name and address information into your microcomputer and storing it for future use. My advice is to set your price to cover wear and tear on your machine and the time you spend entering data. For instance, I charge 5 cents a label for data entry.

With DATBAS and a list of addresses where street name, city, state and ZIP code change infrequently (addressee and street number change each time), Karen, my cohort, can average 100 labels an hour ($5 an hour). This fee is well below that charged by larger firms, which may run as much as 16 cents per label. You can charge less, since your overhead is much lower than theirs.

Next, you can charge for updates, corrections and even sorting the address file to customer specification. For corrections and update, I charge the data entry fee. I do sorts for free. However, since my typical customer doesn't use bulk mail, I've had few requests to sort address files into any particular order. Deletions from the address file are done for free. This encourages clients to purge their files of outdated and unneeded records.

The label charge, for the labels themselves, is where the money is

Program continued.

```
280  C$=A$(253,256) : CONVERT C$ TO R : REM MAX RECORDS FILE CAN HOLD
290  W9=Z
300  FOR I=1 TO Z
310    C$=A$(I*2-1,I*2) : CONVERT C$ TO L(I) : REM LENGTH OF INDIVIDUAL SCREEN LABELS
320    C$=A$(119+I*2,120+I*2) : CONVERT C$ TO D(I) : REM LENGTH OF INDIVIDUAL DATA FIELDS
330    NEXT I
340  FOR I=1 TO 2
350    GET (2,E,A$,I)
360    E$(I*256-255,I*256)=A$ : REM BRING IN LABEL TEXT
370    NEXT I
380  GET (2,E,A$,3)
390  FOR I=1 TO Z
400    C$=A$(I*2-1,I*2) : CONVERT C$ TO R(I) : REM CRT ROW ADDRESS FOR SCREEN LABEL
410    C$=A$(119+I*2,120+I*2) : CONVERT C$ TO C(I) : REM CRT COLUMN ADDRESS FOR SCREEN LABEL
420    NEXT I
430  FOR I=1 TO Z : N1(I)=D(I) : NEXT I
440  S1=1
450  GOSUB 2100
460  C1=0
470  D3=0
480  CURSOR 15,0
490  #"Enter label line, position for data field (l,p) ";
500  FOR I1=1 TO Z
510    CURSOR R(I1),C(I1)+L(I1) : REM POSITION CURSOR BEHIND CRT LABEL
520    INPUT1 D$
530    #" "; : REM BLANK OUT LEFTOVER CURSOR PROMPT
540    Z2=D(I1)-LEN(D$) : FOR Q7=1 TO Z2 : #" "; : NEXT Q7
550    D3=D3+D(I1)
560    CURSOR R(I1),C(I1)+L(I1)
570    #" ";
580    IF D$="99,99" THEN EXIT 760
590    IF LEN(D$)=0 THEN 750
600    C1=C1+1
610    B$=D$
620    FOR I=1 TO LEN(D$)
630      IF B$(I,I)="," THEN EXIT 650 : REM FIND "," SEPARATING ADDRESS LABEL LINE AND LINE POSITION
640      NEXT I
650    C$=B$(1,I-1)
660    CONVERT C$ TO L1(C1) : REM SAVE ADDRESS LABEL LINE ADDRESS
670    D1(C1)=I1 : REM SAVE THE SCREEN LABEL NUMBER
680    X=LEN(D$) : REM FIND LENGTH OF USER INPUT
690    C$=B$(I+1,X) : REM CONVERT INFO AFTER "," TO NUMERIC DATA FOR RELATIVE LINE POSITION
700    CONVERT C$ TO T
710    T=T*.01
720    L1(C1)=L1(C1)+T : REM ADD ADDRESS LINE AND RELATIVE LINE POSITON TOGETHER
730    D(C1)=D(I1) : REM STORE DATA FIELD LENGHT ASSOCIATED WITH USER SELECTED SCREEN LABEL
740    D4(C1)=D3
750    NEXT I1
760  Z=C1 : F0=0
770  FOR I=1 TO C1 : F0=F0+D(I) : NEXT I
780  REM F0 TOTAL LENGTH OF FIELDS SELECTED
790  REM SORT THE USER INPUT INTO ORDER SO THAT ADDRESS LABELS WILL COME OUT AS SPECIFIED.
800  FOR I=1 TO Z-1
810    FOR J=I+1 TO Z
820      IF L1(I)<=L1(J) THEN 920
830      T=L1(I) : T1=D1(I)
840      T2=D(I)
850      T3=D4(I)
860      L1(I)=L1(J) : D1(I)=D1(J)
870      D(I)=D(J)
880      D4(I)=D4(J)
890      L1(J)=T : D1(J)=T1
900      D(J)=T2
910      D4(J)=T3
920      NEXT J
930    NEXT I
940  REM CALCULATE THE BEGINNING AND ENDING POSITION FOR EACH LINE IN ADDRESS LABEL
950  REM CALCULATE THE NUMBER OF LINES IN THE LABEL -- 10 MAX
960  REM          D5  BEGINNING POSITION OF ADDRESS LABEL LINE
970  REM          D6  ENDING POSITION OF ADDRESS LABEL LINE
980  REM          Y9  KEEP TRACK OF NUMBER ADDRESS LABLE LINES
990  REM          C2  NUMBER OF LINES IN ADDRESS LABEL
1000  D5=1 : D6=0 : Y9=1
1010  FOR I=1 TO Z
1020    D6=D6+D(I)
1030    IF INT(L1(I))<>INT(L1(I+1)) THEN 1050
1040    GOTO 1070
1050    D5(Y9)=D5 : D5=D6+1
1060    D6(Y9)=D6 : Y9=Y9+1
1070    NEXT I
1080  C2=1
1090  FOR I=2 TO Z
1100    IF INT(L1(I-1))<>INT(L1(I)) THEN C2=C2+1
1110    NEXT I
1120  #""
1130  INPUT "Selection criteria to be used (Y/N) ",C$
1140  IF C$="Y" OR C$="y" THEN S9=1 : GOSUB 1830
1150  #""
1160  INPUT "Enter number of labels per line ",L8
1170  INPUT "Enter number of lines between labels ",L6
1180  FOR I=1 TO L8
1190    #"For label";%3I;I;" ";
1200    INPUT "enter printer column label is to begin ",T(I)
1210    NEXT I
1220  L5=1
1230  INPUT "Enter beginning record for label output ",B0
1240  #"There are";%5I;Z3;" records currently in the file"
1250  INPUT "Enter ending record for label output ",E0
1260  E1=INT(B0/B1) : E2=B0-E1*B1 : IF E2>0 THEN E1=E1+1
1270  E1=E1+4 : REM BEGINNING DISK BLOCK TO BE READ FOR OUTPUT
1280  IF E2=0 THEN E2=B1
1290  E3=INT(E0/B1) : E4=E0-E3*B1 : IF E4>0 THEN E3=E3+1
```

More

Program continued.

```
1300  E3=E3+5 : Z0=B0 : REM ENDING DISK BLOCK TO BE READ FOR OUTPUT
1310  E$=G$+F$
1320  FOR I=E1 TO E3
1330    GET (2,E,A$,I)
1340    FOR J=E2 TO B1
1350      IF Z0=E0+1 THEN 1490
1360      D9=1 : D4=0
1370      IF S9=0 THEN 1490
1380      IF S7=1 THEN 1450
1390      D$=F$
1400      FOR W=1 TO S8
1410        D$(D2(W)-D3(W)+1,D2(W))=A$(J*D-D+D2(W)-D3(W)+1,J*D-D+D2(W))
1420        NEXT W
1430      IF G$=D$ THEN 1530
1440      GOTO 1620
1450      FOR W=1 TO S8
1460        IF G$(D2(W)-D3(W)+1,D2(W))=A$(J*D-D+D2(W)-D3(W)+1,J*D-D+D2(W)) THEN EXIT 1490
1470        NEXT W
1480      GOTO 1620
1490      IF Z0=E0+1 THEN L8=L5-1 : GOSUB 1690 : EXIT 1660
1500      D1=0
1510      REM THE CODE TO FOLLOW BUILDS ONE ADDRESS LINE FOR ALL LABELS ACROSS THE LABEL CARRIER
1520      REM  WHEN L5  INDICATES WHEN LINE IS BUILT FOR ALL LABELS DUMP THE LINE TO THE PRINTER
1530      FOR I1=1 TO Z : REM NOTE Z NOW HOLDS THE NUMBER DATA FIELDS COMPRISING THE ADDRESS LABEL
1540        D1=D4(I1)
1550        D2=D1-D(I1)+1
1560        D4=D4+D(I1)
1570        E$(L5*F0-F0+D9,L5*F0-F0+D4)=A$(J*D-D+D2,J*D-D+D1)
1580        D9=D4+1
1590        NEXT I1
1600      L5=L5+1
1610      IF L5=L8+1 THEN L5=1 : GOSUB 1690
1620      Z0=Z0+1 : REM NUMBER OF LABELS OUTPUT
1630      NEXT J
1640    E2=1
1650    NEXT I
1660  CLOSE (2,E)
1670  #"End of processing"
1680  END
1690  REM PRINT LABELS
1700  OPEN (PRINTER,E)
1710  D1=0
1720  S1=0
1730  I2=1
1740  FOR I1=1 TO C2 : REM  C2 HOLDS NUMBER OF ADDRESS LABEL LINES
1750    #" "
1760    FOR I3=1 TO L8 : REM  L8 HOLDS NUMBER OF LABELS ACROSS LABEL CARRIER
1770      #TAB(T(I3));E$(I3*F0-F0+D5(I1),I3*F0-F0+D6(I1));
1780      NEXT I3
1790    NEXT I1
1800  FOR L9=1 TO L6 : #" " : NEXT L9
1810  CLOSE (PRINTER,E)
1820  RETURN
1830  REM SELECTION CRITERIA ROUTINE
1840  #""
1850  #" Note: Up to 3 data fields can be used to limit records output"
1860  #" "
1870  INPUT "Enter number of selection criteria ",S8
1880  IF S8<1 OR S8>3 THEN S9=0 : RETURN
1890  INPUT "Must all selection criteria be met (Y/N) ",C$
1900  IF C$="N" OR C$="n" THEN S7=1
1910  GOSUB 460
1920  CURSOR 15,0
1930  #"Enter selection criteria at '?' above ";
1940  S6=1 : D1=0 : D7=0
1950  FOR I=1 TO W9
1960    CURSOR R(I),C(I)+L(I)
1970    INPUT1 D$
1980    Z2=N1(I)-LEN(D$) : FOR I2=1 TO Z2 : #" "; : NEXT I2
1990    D1=D1+N1(I)
2000    #" ";
2010    CURSOR R(I),C(I)+L(I) : #" ";
2020    IF LEN(D$)=0 THEN 2070
2030    D3(S6)=N1(I) : D2(S6)=D1 : D6=N1(I)-1
2040    G$(D1-D6,D1)=D$
2050    S6=S6+1
2060    IF S6=S8+1 THEN EXIT 2090
2070    NEXT I
2080  IF S6<S8+1 THEN U9=1 : GOTO 1830
2090  RETURN
2100  REM SCREEN PRINT ROUTINE
2110  #"" : REM THIS WILL CLEAR SCREEN AND HOME CURSOR
2120  D1=0
2130  J1=1
2140  FOR I1=1 TO W9
2150    CURSOR R(I1),C(I1)
2160    K1=J1+L(I1)-1
2170    #E$(J1,K1);
2180    #" ";
2190    J1=K1+1
2200    D1=D1+N1(I1)
2210    IF S1=1 THEN 2270
2220    FOR M1=1 TO N1(I1)
2230      #"x";
2240      NEXT M1
2250    GOTO 2270
2260    #A$(J*D-D+(D1-(D(I1)-1)),J*D-D+D1);
2270    NEXT I1
2280  S1=0
2290  RETURN
```

made. I charge 3 cents per label. One firm I know charges 2.7 cents per label per thousand. If you buy your labels in large quantities, your label cost will be around .02 cents per label. Once label data is entered, you incur only label costs—the rest is gravy.

If you really become involved in the mailing business, you can offer envelope services. This entails collating and folding materials to be mailed, stuffing envelopes, affixing postage and actually mailing materials.

Finally, there is a delivery charge for getting the finished product to your customer. To remain competitive, about the only thing you can do here is recover out-of-pocket costs.

The Program

The DATBAS programs mentioned earlier make Labels a relatively small program. They relieve the Labels program from file creation and update operations. Labels' only function is to output selected portions of DATBAS files to labels.

Labels first requests the name and drive location of the file requested. Next, a data entry screen is presented, and the program asks you to enter the line and position on the line of each data field to be included in the output label. (If the data field is not to be output, you input a RETURN or ENTER.) A prompting "?" will move through the CRT display until all data fields in the file are exhausted.

Next, you are asked if selection criteria to limit output are to be specified. If so, you enter "Y." You can specify up to three criteria. As with the data specification process phase described previously, you are presented with a CRT display of the data fields in the file. You enter the selection criteria in response to the prompting "?". If a particular data field is not to serve as a selection criterion, you input a RETURN or ENTER. When you have input the previously specified number of criteria, a request is made if "all" or "any one" criterion needs to be met by the record for its output to labels.

Next, you make a request to specify the number of labels across the page or label carrier, the number of print lines between labels and the beginning print position for each label on the carrier. The latter process is similar to setting the tabs on a typewriter. With the entry of this information, you are asked to specify the range of

records in the data file to be read for possible label output. Labels are then output to the system printer.

The flexibility of Labels is shown in Listing 1. One, two, three and four labels across are presented. Of particular interest is the way data field orders can be varied among the label sets.

The program listing for Labels is included. Variable descriptions and remark statements have been included to aid those who might want to use Labels for their own purposes.■

```
FOREST E. MYERS
5114 GARNETT ST
SHAWNEE, KS.   66203

FOREST E. MYERS          CIPHER COMPUTER CO.
5114 GARNETT ST          PO BOX 3508
SHAWNEE, KS.   66203     SHAWNEE, KS.   66203

FOREST E. MYERS          CIPHER COMPUTER CO.      RW SALES
5114 GARNETT ST          PO BOX 3508              PO BOX XXXX
SHAWNEE, KS.   66203     SHAWNEE, KS.   66203     KANSAS CITY, MO64198

FOREST E. MYERS          CIPHER COMPUTER CO.      RW SALES                 WILCOX PROPERTIES
5114 GARNETT ST          PO BOX 3508              PO BOX XXXX              132 BIG BUCKS AVE.
SHAWNEE, KS.   66203     SHAWNEE, KS.   66203     KANSAS CITY, MO64198     BELTON, MO.    64111
```

Listing 1.

Here are some guidelines to help you choose the right computer for your business.

Microcomputer Selection And Implementation

By Bruce T. Pace

In our work as consultants, we've helped many small-businessmen and professionals travel the road to microcomputer system selection and implementation. These are people who, although lacking technical backgrounds, recognize the microcomputer potential, but are unprepared for the onslaught of technical specifications, buzzwords and the variety of products and vendors.

The need for vendor-independent consulting has naturally arisen from a marketplace which lacks true standardization and occasionally masks incompetence with mystique. The intent of this article is to remove the computer mystique and present guidelines to more clearly define the computer selection process.

A Question of Definition

A problem immediately arises when defining a computer system. For some, a given configuration of hardware constitutes a computer system. For many, it is the integration of hardware and software. But those in business who have purchased hardware, software and two days of training will often disagree.

To be used effectively as a tool, the computer system must provide for all aspects of its use. It must produce better than the previous business procedures. To meet the expectations of business, the definition of a computer system should include:

- Hardware
- Software
- Maintenance
- Training
- Documentation
- Human engineering
- Contingency planning

Bruce T. Pace is the president of Microsense Corporation, 7916 Grow Lane, Houston, TX 77040.

Maintenance

The functional system entails much more than hardware and software. Without dependable maintenance, an entire office can come to a devastating standstill. Consideration must be given to repair turnaround time, inventory of a vendor's in-house parts and technical competence.

Although costs are sometimes slightly higher, purchasing the basic software and hardware package from a local source often results in preferred customer status on maintenance and ordering. For the business that is highly dependent upon its computer processing, a maintenance contract is advisable. You may arrange to buy a kit of crucial components for backup.

Training

Since maintenance is critical to continued processing, training is crucial to efficient computer use. Regardless of its potential, the computer will not improve business processing if the staff isn't well-trained. Part of the training should be conducted at the user's site to ease the adjustment to different methods and terminology. The better the training, the less employees will resist change. If your vendor has instruction available, it is advisable to supplement on-the-job training with such a course. This helps counter any problems that may result from word-of-mouth training.

Documentation

Documentation is probably the least-liked and most ignored aspect of computer processing. But without adequate documentation, a computer system can become partly or fully obsolete long before its normal life expectancy.

For most changes you make to your system in the future, you will need the manuals provided by the hardware and software vendors. And if someone alters your system, hang on to that invoice until you are sure the change is clearly documented. If he moves and no one is sure what was done, you may have to go back to square one.

Business procedures established for computer processing must be well-documented. It serves as a ready reference for personnel, provides for problems and reruns and greatly facilitates personnel training. Thorough, current documentation of daily operations and procedures eliminates the problem of any personnel leaving with all of your established procedures in his head.

Human Engineering

The new computer normally involves a change in the methods people establish to perform their tasks. The system should be matched to its environment. The result is increased efficiency and decreased human resistance. The staff should always take part in configuring a system because, simply, they have to live with it. De-

termine the best method for disk use and filing. Consider the keyboard height, ribbon changing, disk care and so on.

Contingency Planning

When a complete system is in place, it's time to consider the "what if" possibilities. What if a fire destroys your computer area? The statistics indicate that the loss of critical data causes a frightening percentage of businesses to close their doors. Or what if an irate employee removes all of your disk storage?

Contingency planning includes data insurance, which is at least as valuable as fire/theft insurance. The insurance company may pay for a new machine and receivables, but it cannot reproduce your data.

First, you should schedule regular backups of computer runs as frequently as you need to. Floppy disks can be corrupted, and vendor or accounting files don't re-create easily.

Second, in case of disaster, keep a copy of critical data, procedures documentation, manuals and source listings in a secure area. A fireproof, locking vault is one method, but storing this material at a different, secure location provides the best protection. And, remember, your contingency materials must be kept up-to-date.

Each potential computer user has different needs. But the above considerations are always important. The entire system must work if the user is to be satisfied.

One Approach

This approach is not the final word in consulting but, rather, an outline of the key areas potential users should consider. It is based on management-level procedures and controls analysis my company has done for larger firms.

When I'm contacted, I present a mini-tutorial to the client and find out what his requirements are. Often, he already has one or two specific packages under consideration. My tutorial includes a lay interpretation of many terms and concepts found in computer processing. Technical specifications are pretty useless if the client doesn't understand such concepts as memory, I/O and software.

I explain to the client what a complete system comprises, and offer him some criteria for selection. We then discuss some of the most important considerations: Is the system cost-effective? Is it compatible or semistandard? Can it be expanded when the business grows? Is maintenance locally available? What is the company background of the vendor? We then order any information not in our files.

Once we've organized the information, we schedule the candidate computer vendors for demonstrations. The people who are initially responsible for the system's operation attend; they are excellent sources for feedback on a workable physical configuration, terminal and printer output quality and user functions. Once we narrow down the choices, we develop alternate configurations for the work station. The user should not discover at installation time that he has no room for interface plugs.

Because a work station is a component of the system as a whole, carpentry, electrical and cabinet costs should be included in the final estimate.

A vendor's price quote usually covers hardware, software and maintenance (if available) prices. But the user needs to consider other expenses which, although more subtle, are very real. They include ac work, programming, manuals, diagnostics, disks, ribbons, print wheels, paper, accessories, training, file creation and initialization, documentation and procedures development.

After installation, other costs arise. For instance, as the company's awareness of the computer's potential grows, so does its need for new software and peripherals. Firms often start out with a special software package for such concerns as word processing and real estate, and evolve to where the computer is supporting all of their accounting.

Patience and education are the critical factors in the selection process. For most firms, a computer is a major investment. Better to take an extra month selecting the system than to be harassed by regret after the dollars are committed.

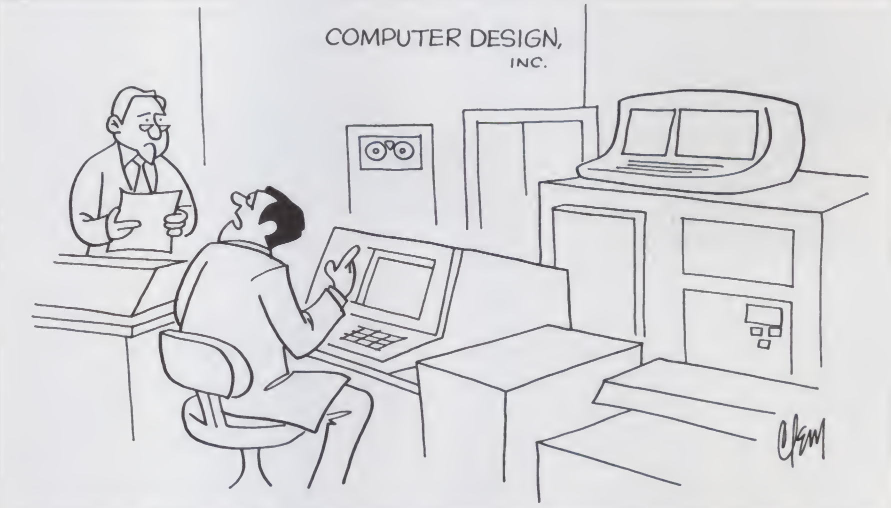
Installation and Training

I'm present at installation to make sure the work station is configured as planned. If the vendor offers installation, we use their experience. Otherwise, I and a technician do it.

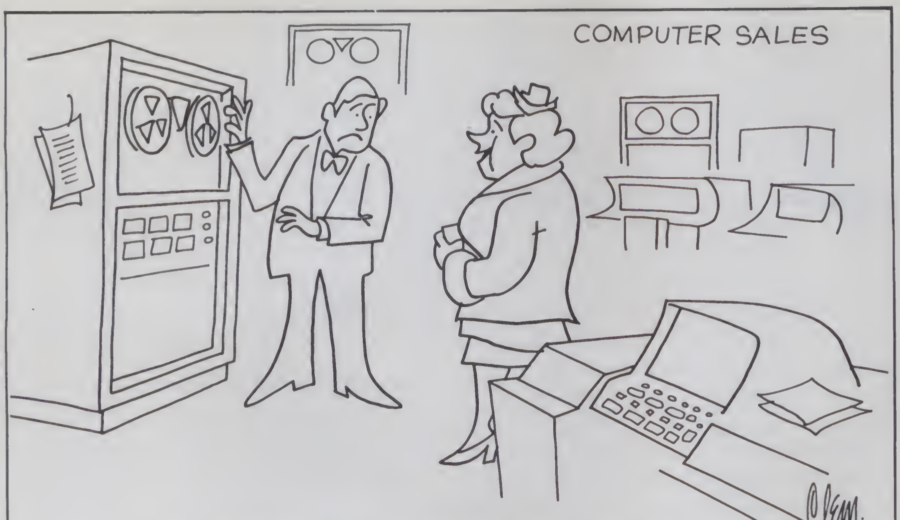
I don't consider the machine installed until it is functional, the software is loaded and tested, all manuals are on site and adequate supplies are available. In some cases, such as word-processing applications, my client will use the system for a period of time, and then return it to the vendor to define the most-needed user function keys.

The training process can be overwhelming for some people if they need to learn the concepts and commands in one training session. It is helpful at this stage to have technical assistance available.

While we are answering questions and clearing up misconceptions, we begin establishing processing procedures. These address time scheduling, run instructions, problem correction, file maintenance and system



"This computer has everything. . . high recording resolution, remote processing, data editing and validation. . . the only thing is, it's not compatible with any other communications media."



"I like it, but do you have one that includes a microwave oven?"

care. Generally, most of the confusion that may arise at the outset is eliminated during the initialization and file creation phase.

While documentation should be outlined during the entire process, now is the time when thorough documentation is a requirement. It is often helpful to condense an operating manual into a quick-reference guide for the most-used commands, whether in a notebook or in a help file on the system. I suggest that the user keep a logbook of malfunctions, problems or repairs to provide a history of system performance. The operations procedures, if well documented, will reduce future training costs and lessen the impact of personnel turnover.

Any modifications made to a standard package must be carefully documented for future reference.

I recommend to clients that they keep the original and new listings for software and the original and new schematics for hardware. If diagnostics are available, they should be scheduled regularly.

After the Sale

After beginning normal use of your system, don't forget that conditions change and that the computer is expandable. Every six months to a year, analyze your processing. The work area may have changed, resulting in decreased efficiency. Additional development, new software, enhancements, additional terminals and so on may now be cost-effective. Review the trouble log to spot any trends that would highlight a specific problem area. And remember: If you make changes without updating your documentation, you might do more harm than good in the future.

Conclusions

Buying your first computer system or upgrading your current system is a significant conversion. The very word "conversion" sends shivers through many data-processing managers in their third year of a two-year plan. The small-businessman or professional faces the same problems, the difference being one of scale.

Like the big shops, the key to success lies in planning, patience and adherence to procedures and controls. And, yes, in most cases there is a learning curve for your personnel that must be accounted for. After three to five days of training, a person will not normally be up to full speed and efficiency.

Be realistic about selecting and implementing a computer system. Get some nonaligned technical assistance if possible. We who are a part of the growing number of consultants must accept a professional responsibility for aiding the nontechnical person in investing company funds wisely. The enlarged concept of a system merges well with an organized business approach.

Be not discouraged! Planned carefully, the computer is an important new tool for increased efficiency and growth in your business.■

Call For Manuscripts

Kilobaud Microcomputing is looking for business articles!

Businessmen in all fields are beginning to take notice of the microcomputer. They are eager to know which computers, peripheral equipment and applications software will let them take full advantage of this new tool. What knowledge do you have to share?

Here are the kinds of articles that we want you to write for us:

- Are you a businessman with a system up and running? We want to know how it works. What were your expectations? Have they been fulfilled? Did you find the software that you wanted? What problems have you had? How did you overcome them? What recommendations do you have for other businessmen?
- We want reviews from a businessman's perspective of specific hardware and software. If you've recently bought a new product and want to tell others how great—or poor—it is, *Microcomputing* will provide you with a forum.
- What programs have you written to meet your specific needs? Perhaps another businessman can use them, too. Even if he can't, your program may serve as a springboard for other ideas.
- Perhaps you aren't using your micro for business, but know a company that is. Trot on down with your pencil and notebook, and find out what they're up to. While they might not have the time to write up their experiences, they might be more than willing to tell somebody else about them. And an outside observer will often be able to see things with a unique and valuable perspective.

Don't worry if you're not a professional writer. That's what we editors are here for. And we'll be more than happy to send you a copy of our writer's guidelines.

Send your manuscripts and correspondence to:
Kilobaud Microcomputing
Pine St.
Peterborough, NH 03458

♄ ♀ ♂ ♃ ♇ ☉ ♆

Astrology and The Microcomputer

By J. Lee Lehman

Astrology has entered the computer age. Some 3½ percent of astrologers today own micros, and the figure is rising. Another 25 percent use either astrological computing services or their friends' micros. Companies like Astro-Graphic Services are offering increasingly sophisticated software for such computers as the Apple, PET and TRS-80.

Astrologers are finding that computers can be invaluable aids in a number of areas. They save time, make calculations easier for astrologers who have trouble with math and can pave the way for large statistical research projects. Most importantly, they free the astrologer to concentrate on chart interpretation.

How is it that astrology can be compatible with a space-age technology like computers? First, astrology is *not* the daily newspaper horoscope; we would have been better off if those had gone out with the Middle Ages. Rather, it is the study of the locations of the sun, moon, and the planets and some of the asteroids at the time of someone's—or something's—birth. An astrologer needs to know the exact birth time, and accurate calculations are critical to accurate forecasts and interpretations. The computer is an ideal means of achieving this goal.

The Basis of Astrology

The premise of astrology is that there is a connection between planetary locations at birth and the way in which an individual's life unfolds.

Why such a connection exists is unknown, but it has been established by such writers as Dr. Michel Gauquelin. Gauquelin studied the birth charts of thousands of sports figures, physicians and scientists, soldiers, artists, actors, journalists, politicians, business executives, writers and musicians.

He found that the locations of Jupiter, Mars, Saturn and the moon were not random for those who were more successful.

The calculated probability of his results occurring by chance runs from 1 in 100 for journalists to 1 in 5 million for sports champions. When was the last time you saw a behavioral study with a chance probability of 1 in 5 million?

While Gauquelin's work does not by any means prove that astrology works, he has established that there is something to planetary effects. And that is where your friendly local astrologer comes in. Any decent professional astrologer should be able to give you a fairly detailed personality description, and tell you what's doing with your mid-life crisis, or whether you should consider changing careers, or when the investment climate is good for you.

When I do a reading for a client, I generally calculate three separate horoscopes or charts: the birth (natal) chart and two supplemental ones which give me a handle on what the client is going through that year. These calculations could easily take several hours by hand, 45 minutes by calculator and about 20 minutes by microcomputer. When I do it on the computer, that 20 minutes gives me more information than I would have gotten via calculator, because it's just too much trouble to do some calculations routinely.

Computerization Arrives

Computerization has been creeping up on astrologers gradually. First, the ephemerides—books that list daily planetary locations—began to be calculated by computer. Then, in 1971, Neil Michelsen, then with IBM, founded Astro Computing Services (ACS). ACS, which now uses a Prime 350 minicomputer, rapidly built up a reputation for both accurate charts and fast service. They also produce ephemerides and other reference books which have become the most trustworthy in the field. An astrologer could get the three above charts calculated for about $7.

A number of other computer services followed, with varying degrees of accuracy. (I've seen some printouts from the early period which were virtually useless.) Of comparable accuracy, and somewhat lower

Dr. J. Lee Lehman resides at 325 E. 21st St., #25, New York, NY 10010.

price, is Astro-Graphics Services, which, unlike ACS, was founded by astrologers rather than computerists.

However, $5–$7 per client mounts up if you have many clients, and sometimes even waiting a week is too much. Enter James Neely and Michael Erlewine. Neely took the astronomical equations for planetary motion—equations that had been developed for mainframe computers in which memory size was not limiting—and worked out the algorithms that allowed acceptable accuracy using 8K, and then 16K, of memory. Erlewine published these programs in *MATRIX Magazine* and then developed MATRIX Software, which *provides these progams* for the Apple II, PET and TRS-80 Level II. (The 8K programs are noticeably less accurate for the positions of the outer planets than the 16K ones.)

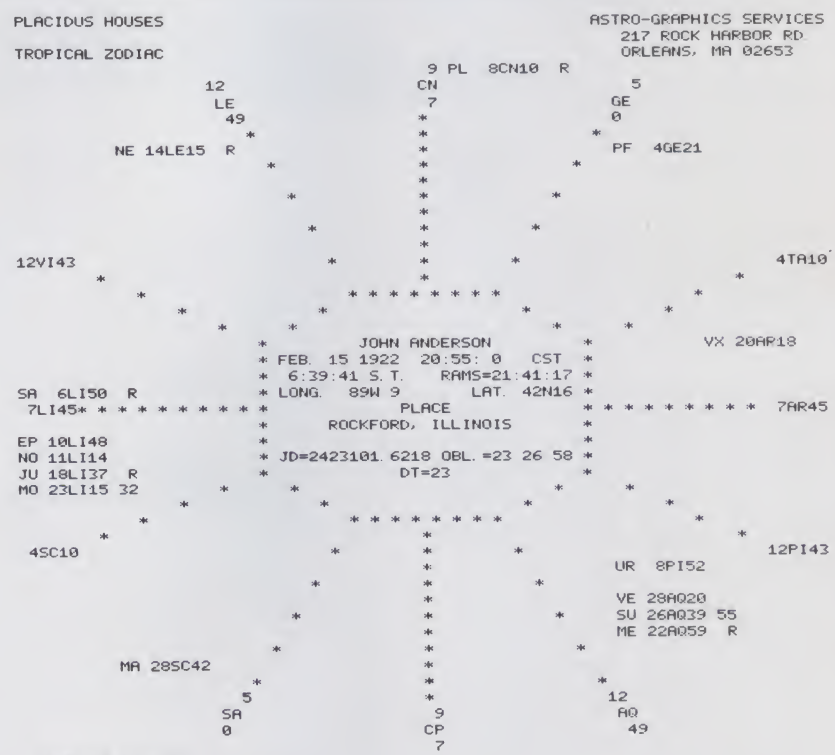
These programs are primarily limited to the 20th century, although more extensive ranges are becoming available. Erlewine has recently published a summary of this programming work—including a full listing of the basic 8K calculations—in his book *Manual of Computer Programming for Astrologers.*

Besides the three micros mentioned above, there is one additonal entrant—Digicomp's DR-70. The DR-70 is an astrologically dedicated computer with 32K of ROM programming. Until recently, it offered the largest number of software options. Consisting of two compact units, the CPU and printer, it is far more portable than the other micros. However, it has the rather stiff retail price of $3000. And it doesn't play chess.

Sales on the Rise

Within the last year, the sale of micros to astrologers has shot up. Henry Weingarten of Astrology Services International in New York City (ASI sells the Apple II, PET, PMC-80 and DR-70, and software for all the micros) estimates that about 3–5 percent of astrologers own their own computers, and 25 percent use either the astrological computing services or their friends' micros. He expects that number to double within the next year, and by 1985 he anticipates that at least 30 percent will own micros and 95 percent will use computers.

Right now, the DR-70 represents about 15–20 percent of the market. The rest is about evenly distributed between TRS-80, Apple and PET, although the latter two have an advantage. Michael Erlewine is a PET dealer and actively promotes it over the other micros. Furthermore, astrology centers like ASI can only sell Apples or PETs; Radio Shack is ignorant of the astrology programs available, and the astrologer has to get by without any encouragement from the astrological sources.



EP=EAST POINT
PF=PARS FORTUNA
VX=VERTEX

PLANET	GEO. LONG.	TRAVEL	GEO. LAT.	RT. ASC.	DECLIN.	HEL. LONG.	HEL. LAT.
SUN-EARTH	26 AQ 39 55	1 1	+0 0	328 54	-12 38	26 LE 40	+0 0
MOON	23 LI 15 32	14 21	+1 13	201 59	-7 54	26 LE 40	+0 0
MERCURY	22 AQ 59 R	-1 8	+3 43	324 6	-10 21	3 VI 23	+6 44
VENUS	28 AQ 20	+1 15	-1 23	330 59	-13 22	0 PI 36	-3 16
MARS	28 SC 42	+0 31	+1 4	236 42	-18 50	20 LI 59	+0 52
JUPITER	18 LI 37 R	-0 8	+1 29	197 45	-5 55	10 LI 25	+1 18
SATURN	6 LI 50 R	-0 5	+2 33	187 16	-0 22	2 LI 59	+2 20
URANUS	8 PI 52	+0 3	-0 44	340 45	-8 56	9 PI 27	-0 46
NEPTUNE	14 LE 15 R	-0 2	+0 7	136 45	+16 40	14 LE 39	+0 7
PLUTO	8 CN 10 R	-0 1	-3 9	98 41	+20 3	9 CN 9	-3 6
N. NODE	11 LI 14	-0 3	+0 0	190 19	-4 27	0 0	+0 0

AGS SOFTWARE

A portion of the printout of AGS Software. A birth chart is read starting at the nine o'clock position, or rising sign, in this case 7 degrees, 45 minutes of Libra. The planets are found in the 12 sectors of the chart known as houses, which are counted counterclockwise from the rising sign, which delimits the beginning of the first house. Anderson's Saturn, ascendant, North Node and East Point are very close to the degree of this year's Jupiter-Saturn conjunctions, which have been the astrological indicators of the phenomenon whereby presidents elected in even decade years have died in office. Carter also has planets in this region. Reagan has been cagy about releasing his birth data, so his exact chart is not known.

The author on the trail of a hot horoscope. The graphic on the screen is part of the MATRIX Software Natal Chart Program for the TRS-80. (Photo by Tee A. Corinne.)

According to Weingarten, "Fifty percent of astrologers buy computers for computers, and astrology is the extra reason, or excuse, for buying. Fifty percent are buying because they can't do the math. The ones who are not mathematically sophisticated will often buy the dedicated DR-70, or a very simple computer, 16K. The people who are into more sophisticated uses will buy more memory, a printer and disks."

The computer sophisticates are finding themselves with more options. In June 1980, AGS made their disk-based software available for the TRS-80 and Apple II. These disk programs feature better graphics and more precision, which in this case means that they are accurate for both 19th and 20th century births. (The MATRIX 16K programs lose out on the accuracy of the positions of the planets Jupiter through Pluto if you go back much earlier than 1900.)

I asked Robert Hand of AGS if he thought they were pricing themselves out of the market, since those of us who build our systems by upgrading would already have the MATRIX programs. He replied, "Our object is to allow people the option of buying computer service quality software for micros; we're just making our software available in addition to our chart calculation."

The disk-based systems promise a considerable savings in time, since quite a number of different modules can be run off the main natal calculations. (Unfortunately, the MATRIX programs are all separate, so that the birth chart has to be recalculated along with each different kind of supplemental chart, a considerable timewaster.) Hand added that AGS will soon market astronomy programs as well, including planetary observation (angles, declinations and times) and fixed stars.

Looking Ahead

What's ahead for astrology in the computer age? Computers are certainly *not* putting astrologers out of work; if anything, they are making us work harder.

Previously, the most limiting factor in providing a reading was the amount of time one had to devote to the tedious calculations necessary to provide answers to specific questions. For instance, suppose a client is wondering whether he or she would be better off moving to a new location. By calculating one supplemental chart (that's seven minutes with good calculator technique), the astrologer can determine whether the client is likely to move during the course of the next year. If the answer is yes, then the next question is where to move to: the next town, across country, north or south?

Ideally, the astrologer calculates either a relocation chart for each city the client is considering, or at least the predominant planetary influences for each location. (Yet another facet of computerized astrology to provide a world view of the effects of location was developed by Jim Lewis. His Astro*Carto*Graphy printouts provide a map of the world and the individual's planetary strengths for all locations.) This is going to require a few minutes of time per location. It is apparent that without a computer it becomes difficult to give an immediate answer to a client during the course of a reading.

Other kinds of questions can also be answered almost immediately with the computer, provided the astrologer takes the time to write a few utility programs. I have found that since I got my computer, my readings are both more detailed and more attuned to the issues that concern the client.

Another important change of the computer age is that astrologers are now in a better position to evaluate the many methods of chart interpretation which have been kicking around. Astrology is in a transitional period; a lot of rules are still around from centuries ago. *Computers are* providing the means to test them on large numbers of charts.

This is especially important because most of the major astrological discoveries of the past few centuries—besides learning about the planets and asteroids discovered during this period—have consisted of finding some new calculation to answer a specific question. There are so many different techniques around that few astrologers have had the opportunity to try very many of them on more than one or two charts. As a result, it has been very difficult to sort out which techniques are generally applicable to a large number of situations. Now the limiting factor is the willingness of the astrologer to write new programs.

Computerization is making it feasible to do statistical research on a massive scale. Accordingly, the possibility of building up an astrological data base—of people who have food allergies, or club feet or whatever—is more exciting.

For the layperson, the entry of computers into astrology increases the likelihood that the birth chart is accurate. Computers are freeing the astrologer to do what he or she does best: chart interpretation.

If you want to evaluate the validity of astrology for yourself, consider shelling out the $30–50 to your local

computerized astrologer for a reading. If you like what you hear, you just might find another use for your micro! ■

References

- Astro*Carto*Graphy, PO Box 22293-U, San Francisco, CA 94122.
- Astro Computing Services, PO Box 16297, San Diego, CA 92116.
- Astro-Graphics Services, 217 Rock Harbor Road, Orleans, MA 02653.
- Astrological Services International, 127 Madison Ave., New York, NY 10016.
- *Astrology Now,* Issue #27, "The Computer Boom." Published by Llewellyn Publications, 213 E. 4th St., St. Paul, MN 55101.
- Erlewine, Michael. 1980. *Manual of Computer Programming for Astrologers.* American Federation of Astrologers, Inc., PO Box 22040, Tempe, AZ 85282. 216 pp.
- Gauquelin, Michel. 1978. *Cosmic Influences on Human Behavior.* ASI Publishers (see above): New York. 319 pp.
- MATRIX Software, 1041 N. Main St., Ann Arbor, MI 48104.

René Descartes * Homer * Jeremy Bentham * Plato * John Adams * Pliny the Elder

Galileo Galilei * Sigmund Freud * Michelangelo * Ludwig van Beethoven * Leibnitz * Thomas Hobbes *

Computer\n

1: Device designed to execute a sequence of mathematical operations.

Education for the Home

Beginner's Russian

This package consists of three programs that graphically display the Cyrillic alphabet. The programs are arranged so that you progress from one to the next—building your knowledge as *you* progress. It includes instructions on proper pronunciation of the letters and even an introduction to simple Russian words.
Order No. 0136R $9.95

Everyday Russian

Everyday Russian will acquaint you with the Russian words relating to: foods, places to eat, everyday signs, and the names of common stores. You will also learn the order of the Cyrillic alphabet. Each of the three divisions of this package will teach you the words and then quiz you on comprehension. You can even practice typing in Russian, using your TRS-80 keyboard as a "Cyrillic typewriter."
Order No. 0137R $9.95

The Russian Disk

Now you can have *both* the Beginner's Russian and Everyday Russian packages on floppy disk! Requires an Expansion Interface with 16K and one disk drive.
Order No. 0212RD $24.95

Teacher

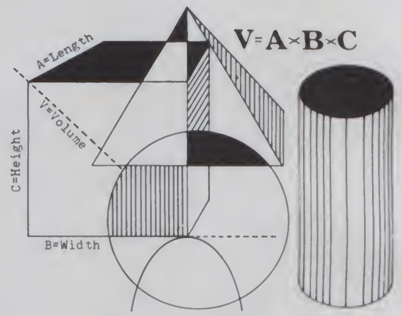
This program allows you to input any number of questions and answers. The computer will prepare tests, give quizes, provide up to three hints per question and even give (optional) graphic rewards for correct answers. Perfect for parents, teachers, or anyone faced with learning a lot of data in a short time.
Order No. 0065R $9.95

Wordwatch

Four programs for budding lexicographers, etymologists, or anyone else who uses words. In WORD RACE, you must choose the proper definitions. Find the misspelled word in HIDE N SPELL. Take a pre-recorded quiz in SPELLING BEE, in which the words are played aloud! Meet variations on proper spelling in SPELLING TUTOR.
Order No. 0111R $7.95

IQ Test

Are you smart enough to buy this package? IQ Test will administer and score an intelligence test in 30 minutes flat! There are three equivalent tests, each consisting of 35 questions, designed to test your general knowledge and problem solving abilities. Most of us claim a "touch of genius"—here's your chance to prove it!
Order No. 0157R $9.95



Archimedes' Apprentice

A tutorial software package that will teach you the formulas used to find the volume of *any* solid object. It covers parallelopipeds (cubes and rectangular solids), prisms, pyramids, cylinders, cones and spheres. It can even quiz you on how well you learned the lesson.
Order No. 0092R $9.95

Video Speed-Reading Trainer

You can increase your reading speed and comprehension. How? By practicing, that's how! This three-part program will flash characters or words on the screen, then you must echo what you saw. You can begin at a relatively slow rate, because the computer will advance your speed automatically as your speed and comprehension increase. It will train you with numbers, letters, words and phrases.
Order No. 0100R $9.95

Typing Teacher

A complete seven-part package that guides you from familiarization with the keyboard, through typing words (and phrases), to mastery of touch typing. Your video monitor becomes a bottomless page for typing practice!
Order No. 0099R $9.95

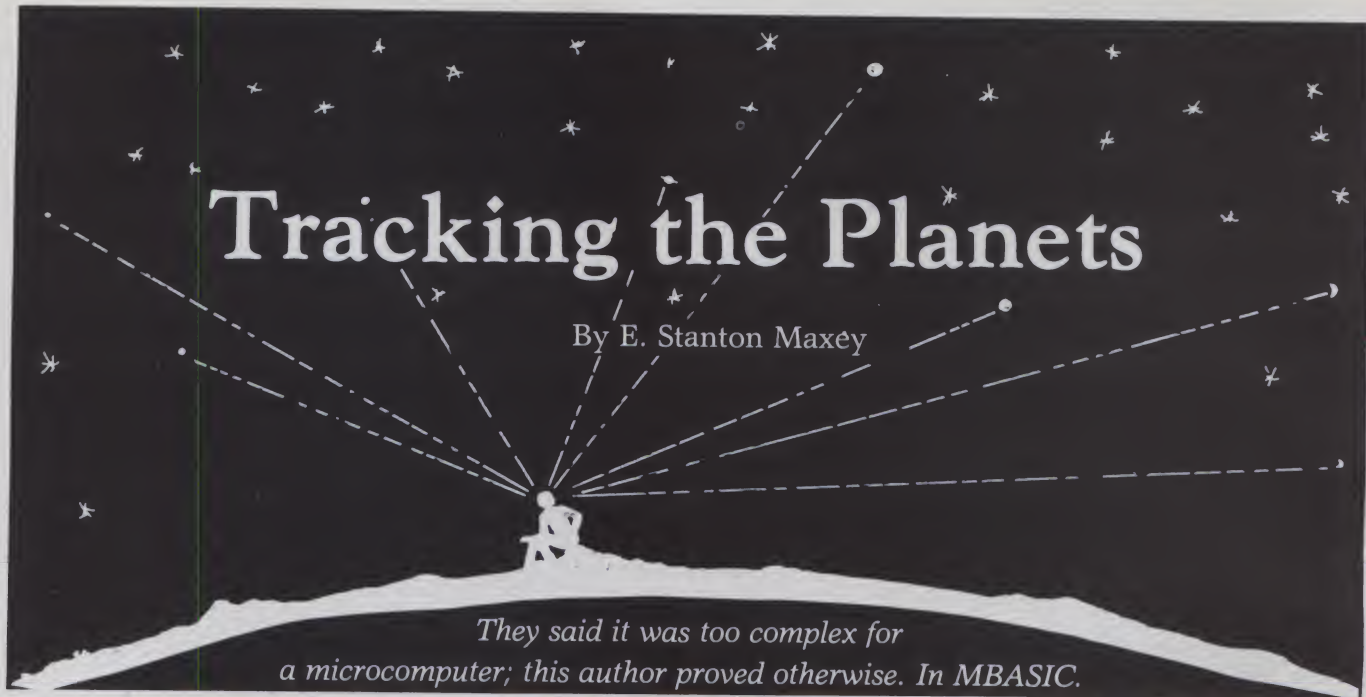
All packages listed are for the TRS-80 Model I Level II; they require 16K of memory and are cassette-based unless otherwise indicated.

Instant Software™ PETERBOROUGH, N.H. 03458 603-924-7296

Rebbe Akiva * John Stuart Mill * Pythagoras * Niccolo Machiavelli * Sun Tzu Wu *

Tracking the Planets

By E. Stanton Maxey



They said it was too complex for a microcomputer; this author proved otherwise. In MBASIC.

Demands for calculating planetary coordinates are increasing. Applications range from sighting telescopes and satellite receiving antennae dishes to the erection of astrological charts.

Elliptical orbital equations can be found in any elementary physics text, but using such equations in calculating planetary coordinates to any reasonable precision is no simple task. The complexities of multiple bodies interacting with each other within their separate orbits is severe. One professor of astronomy and computer technology has said that the task of computing positions for only three orbiting bodies was too complex for microcomputers; that at the very least a minicomputer would be required.

I needed to determine heliocentric longitudes during research into terrestrial magnetic storms. Inquiries at the Science Museum & Planetarium of Palm Beach County, FL, led me to Dr. K. F. Pulkkinen of the U.S. Naval Observatory, Washington, DC. Dr. Pulkkinen and T. C. Van Flandern co-authored *Low Precision Formulae for Planetary Positions.* This paper provides low precision formulae for geocentric and heliocentric positions of the sun, moon and planets for any epoch within 300 years of the present. Because I required only heliocentric data, the program addresses only that requirement. Expansion for geocentric data could readily be achieved.

I used Microsoft's MBASIC program, but it can be adapted to any specific BASIC with minimal editing.

The program requires ephemeris time (equivalent to Universal time for this application), time in Julian centuries and the Julian date. The Julian date (IJD) is calculated in lines 2800–2870, Universal time (ITT) in

Program listing in Microsoft MBASIC. The program examples are for planets 0 through 8 (Mercury through Jupiter) for the test date of zero hours on June 28, 1969. The precision of result is determinable by comparing the program yield to λ values found in Table 3 of the Van Flandern and Pulkkinen paper. The heliocentric longitude of planet 2 (Earth) is the reciprocal of the Sun's longitude. The program is an operational skeleton which can be readily modified by users to meet individual requirements. The author can provide a copy on punch tape for $25.

```
10 'This program is adapted from...
   "Low-Precision Formulae for Planetary Positions"

   by T.C Van Flandern and K.F.Pulkkinen
   US Naval Observatory, Washington
20 'The Astrophysical Journal Supplement Series,
   41:391-411,1979 November
30 '

Program written by
   E. Stanton Maxey, M.D.
   921 E. Ocean Blvd.
   Stuart, Fla.  33494
40 PRINT"This is Planet."
50 DEFDBL I,B
60 DEFINT V,A,C
70 DEFSTR P
80 DIM B(33),BA(33),BB(33),P(8,50)
90 BP2=6.283185307#'Pi*2
100 P="##.##":P1="##":P2="###"
110 BP3=1296000#'no. arc sec in circle
120 '

   Fundamental Arguments
130 BA(7)=.779072#:BB(7)=.00273790931#'Ls
140 BA(8)=VAL(".993126"):BB(8)=.0027377785#'Gs
150 BA(9)=.700695#:BB(9)=.011367714#'L1
160 BA(10)=.485541#:BB(10)=.01136759566#'G1
170 BA(11)=.566441#:BB(11)=.01136762384#'F1
180 BA(12)=VAL(".505498"):BB(12)=VAL(".00445046867")'L2
190 BA(13)=.140023#:BB(13)=.00445036173#'G2
200 BA(14)=.292498#:BB(14)=.00445040017#'F2
210 BA(15)=VAL(".987353"):BB(15)=.00145575328#'L4
220 BA(16)=VAL(".053856"):BB(16)=.00145561327#'G4
230 BA(17)=VAL(".849694"):BB(17)=.00145569465#'F4
240 BA(18)=.089608#:BB(18)=.00023080893#'L5
250 BA(19)=VAL(".056531"):BB(19)=BB(18)'G5
260 BA(20)=VAL(".814794"):BB(20)=BB(18)'F5
270 BA(21)=.133295#:BB(21)=.00009294371#'L6
```

More

Dr. E. Stanton Maxey is with the Ocean Boulevard Medical Center, 921 East Ocean Blvd., Stuart, FL 33494.

line 2880 and time in Julian centuries (IT) in line 2890. (IT) is the only polynomial variable needed for the program.

Line 2920 calculates values for the fundamental arguments using Universal time (ITT) and the arguments defined in lines 130–380. These arguments are taken directly from the Pulkkinen paper. Note that val (".821218") is used instead of (.821218#), since the B variable is defined as double precision and MBASIC likes to print these values in 16 digits. The val function saves some memory here.

The expression for heliocentric longitude—B(V)—is calculated in lines 2920–2930 and lines 3080–3280. For example, the first four terms of the series for the longitude of Jupiter are computed as follows:

$$
\begin{aligned}
B(V) - .089608 + .00023080893 \cdot ITT = {} & 19934 \cdot \sin(.056531 + .00023080893 \cdot ITT) \\
& + 5023 \cdot IT \\
& + 2511 \\
& + 1093 \cdot COS\,(2 \cdot (.056531 + .0023080893 \times ITT) - 5 \cdot (.0882987 + .00009294371 \cdot ITT)).
\end{aligned}
$$

Arcseconds are converted to radians in line 3120. String arrays are extensively employed, since their characteristics allow incorporation of the PLON (periodic terms) in an easily accessible format. The zero string, e.g., P(8,0) = "3222" in line 2570, conveys the numbers of the fundamental arguments; in this case argument 32 is found in line 370 and argument 33 in line 380. The lengths of variables P(0,0) through P(8,0) enable dynamic computation (in lines 3100–3130), depending on the number of fundamental arguments used.

Lines 3320–3390 convert arcseconds to degrees, minutes and seconds, which are then printed out as such. I consistently got precision of less than 1 minute of arc as checked by the American Ephemeris and Nautical Almanac. The exception was Pluto, which was up to 5 minutes of arc in error. This is understandable, since Van Flandern and Pulkkinen claim only 15 minutes precision for Pluto.

The program as written will require 13,700 bytes of memory after BASIC is loaded. Users short of memory can separate the program into two portions and store all string PLON arrays in magnetic memory *with a shorter* first program. The string variables can then be loaded from disk or cassette as part of a much shorter operating second program. ■

Listing continued.

```
280 BA(22)=.882987#:BB(22)=BB(21)'G6
290 BA(23)=VAL(".821218"):BB(23)=BB(21)'F6
300 BA(24)=VAL(".870169"):BB(24)=VAL(".00003269438")'L7
310 BA(25)=.400589#:BB(25)=BB(24)'G7
320 BA(26)=VAL(".664614"):BB(26)=.00003265562#'F7
330 BA(27)=.846912#:BB(27)=.00001672092#'L8
340 BA(28)=VAL(".725368"):BB(28)=BB(27)'G8
350 BA(29)=.480856#:BB(29)=.00001663715#'F8
360 BA(31)=VAL(".663854"):BB(31)=.00001115482#'L9
370 BA(32)=.04102#:BB(32)=.00001104864#'G9
380 BA(33)=.357355#:BB(33)=BB(32)
390 P(0,0)="101113"'each 2 digits=no. of fundamental argument
400 '

    PLON coefficients
410 'length of string gives total arguments used
420 '+/- allows string splitting and evaluation
430 '

   Mercury
440 P(0,1)="84378 0S+ 1+ 0+ 0"
450 P(0,2)="10733 0S+ 2+ 0+ 0"
460 P(0,3)="1892 0S+ 3+ 0+ 0"
470 P(0,4)="-646 0S+ 0+ 2+ 0"
480 P(0,5)="381 0S+ 4+ 0+ 0"
490 P(0,6)="-306 0S+ 1- 2+ 0"
500 P(0,7)="-274 0S+ 1+ 2+ 0"
510 P(0,8)="-92 0S+ 2+ 2+ 0"
520 P(0,9)="83 0S+ 5+ 0+ 0"
530 P(0,10)="-28 0S+ 3+ 2+ 0"
540 P(0,11)="25 0S+ 2- 2+ 0"
550 P(0,12)="19 0S+ 6+ 0+ 0"
560 P(0,13)="-9 0S+ 4+ 2+ 0"
570 P(0,14)="8 1S+ 1+ 0+ 0"
580 P(0,15)="7 0C+ 2+ 0- 5"
590 '

    Venus
600 P(1,0)="081314"
610 P(1,1)="2814 0S+ 0+ 1+ 0"
620 P(1,2)="-181 0S+ 0+ 0+ 2"
630 P(1,3)="-20 1S+ 0+ 1+ 0"
640 P(1,4)="12 0S+ 0+ 2+ 0"
650 P(1,5)="-10 0C+ 2- 2+ 0"
660 P(1,6)="7 0C+ 3- 3+ 0"
670 '

    Earth
680 P(2,0)="010708131619"
690 P(2,1)="6910 0S+ 0+ 0+ 1+ 0+ 0+ 0"
700 P(2,2)="72 0S+ 0+ 0+ 2+ 0+ 0+ 0"
710 P(2,3)="-17 1S+ 0+ 0+ 1+ 0+ 0+ 0"
720 P(2,4)="-7 0C+ 0+ 0+ 1+ 0+ 0- 1"
730 P(2,5)="6 0S+ 1- 1+ 0+ 0+ 0+ 0"
740 P(2,6)="5 0S+ 0+ 0+ 4+ 0- 8+ 3"
750 P(2,7)="-5 0C+ 0+ 0+ 2- 2+ 0+ 0"
760 P(2,8)="-4 0S+ 0+ 0+ 1- 1+ 0+ 0"
770 P(2,9)="4 0C+ 0+ 0+ 4+ 0- 8+ 3"
780 P(2,10)="3 0S+ 0+ 0+ 2- 2+ 0+ 0"
790 P(2,11)="-3 0S+ 0+ 0+ 0+ 0+ 0+ 1"
800 P(2,12)="-3 0S+ 0+ 0+ 2+ 0+ 0- 2"
810 '

    Mars
820 P(3,0)="0813161719"
830 P(3,1)="38451 0S+ 0+ 0+ 1+ 0+ 0"
840 P(3,2)="2238 0S+ 0+ 0+ 2+ 0+ 0"
850 P(3,3)="181 0S+ 0+ 0+ 3+ 0+ 0"
860 P(3,4)="-52 0S+ 0+ 0+ 0+ 2+ 0"
870 P(3,5)="37 1S+ 0+ 0+ 1+ 0+ 0"
880 P(3,6)="-22 0C+ 0+ 0+ 1+ 0- 2"
890 P(3,7)="-19 0S+ 0+ 0+ 1+ 0- 1"
900 P(3,8)="17 0C+ 0+ 0+ 1+ 0- 1"
910 P(3,9)="17 0S+ 0+ 0+ 4+ 0+ 0"
920 P(3,10)="-16 0C+ 0+ 0+ 2+ 0- 2"
930 P(3,11)="13 0C+ 1+ 0- 2+ 0+ 0"
940 P(3,12)="-10 0S+ 0+ 0+ 1- 2+ 0"
950 P(3,13)="-10 0S+ 0+ 0+ 1+ 2+ 0"
960 P(3,14)="7 0C+ 1+ 0- 1+ 0+ 0"
970 P(3,15)="-7 0C+ 2+ 0- 3+ 0+ 0"
980 P(3,16)="-5 0S+ 0+ 1- 3+ 0+ 0"
990 P(3,17)="-5 0S+ 1+ 0- 1+ 0+ 0"
1000 P(3,18)="-5 0S+ 1+ 0- 2+ 0+ 0"
1010 P(3,19)="-4 0C+ 2+ 0- 4+ 0+ 0"
1020 P(3,20)="4 1S+ 0+ 0+ 2+ 0+ 0"
1030 P(3,21)="4 0C+ 0+ 0+ 0+ 0+ 1"
1040 P(3,22)="3 0C+ 0+ 1- 3+ 0+ 0"
1050 P(3,23)="3 0S+ 0+ 0+ 2+ 0- 2"
1060 '

     Jupiter
1070 P(4,0)="18192225"
1080 P(4,1)="19934 0S+ 0+ 1+ 0+ 0"
1090 P(4,2)="5023 1C+ 0+ 0+ 0+ 0"
1100 P(4,3)="2511 0C+ 0+ 0+ 0+ 0"
1110 P(4,4)="1093 0C+ 0+ 2- 5+ 0"
```

More

Listing continued.

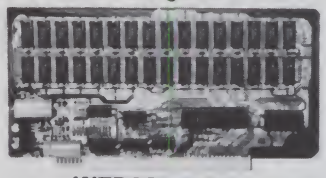
```
1120 P(4,5)="601 0S+ 0+ 2+ 0+ 0"
1130 P(4,6)="-479 0S+ 0+ 2- 5+ 0"
1140 P(4,7)="-185 0S+ 0+ 2- 2+ 0"
1150 P(4,8)="137 0S+ 0+ 3- 5+ 0"
1160 P(4,9)="-131 0S+ 0+ 1- 2+ 0"
1170 P(4,10)="79 0C+ 0+ 1- 1+ 0"
1180 P(4,11)="-76 0C+ 0+ 2- 2+ 0"
1190 P(4,12)="-74 1C+ 0+ 1+ 0+ 0"
1200 P(4,13)="68 1S+ 0+ 1+ 0+ 0"
1210 P(4,14)="66 0C+ 0+ 2- 3+ 0"
1220 P(4,15)="63 0C+ 0+ 3- 5+ 0"
1230 P(4,16)="53 0C+ 0+ 1- 5+ 0"
1240 P(4,17)="49 0S+ 0+ 2- 3+ 0"
1250 P(4,18)="-43 1S+ 0+ 2- 5+ 0"
1260 P(4,19)="-37 0C+ 0+ 1+ 0+ 0"
1270 P(4,20)="25 0S+ 2+ 0+ 0+ 0"
1280 P(4,21)="25 0S+ 0+ 3+ 0+ 0"
1290 P(4,22)="-23 0S+ 0+ 1- 5+ 0"
1300 P(4,23)="-19 1C+ 0+ 2- 5+ 0"
1310 P(4,24)="17 0C+ 0+ 2- 4+ 0"
1320 P(4,25)="17 0C+ 0+ 3- 3+ 0"
1330 P(4,26)="-14 0S+ 0+ 1- 1+ 0"
1340 P(4,27)="-13 0S+ 0+ 3- 4+ 0"
1350 P(4,28)="-9 0C+ 2+ 0+ 0+ 0"
1360 P(4,29)="9 0C+ 0+ 0+ 1+ 0"
1370 P(4,30)="-9 0S+ 0+ 0+ 1+ 0"
1380 P(4,31)="-9 0S+ 0+ 3- 2+ 0"
1390 P(4,32)="9 0S+ 0+ 4- 5+ 0"
1400 P(4,33)="9 0S+ 0+ 2- 6+ 3"
1410 P(4,34)="-8 0C+ 0+ 4-10+ 0"
1420 P(4,35)="7 0C+ 0+ 3- 4+ 0"
1430 P(4,36)="-7 0C+ 0+ 1- 3+ 0"
1440 P(4,37)="-7 0S+ 0+ 4-10+ 0"
1450 P(4,38)="-7 0S+ 0+ 1- 3+ 0"
1460 P(4,39)="6 0C+ 0+ 4- 5+ 0"
1470 P(4,40)="-6 0S+ 0+ 3- 3+ 0"
1480 '

    Saturn
1490 P(5,0)="19212225"
1500 P(5,1)="23045 0S+ 0+ 0+ 1+ 0"
1510 P(5,2)="5014 1C+ 0+ 0+ 0+ 0"
1520 P(5,3)="-2689 0C+ 2+ 0- 5+ 0"
1530 P(5,4)="2507 0C+ 0+ 0+ 0+ 0"
1540 P(5,5)="1177 0S+ 2+ 0- 5+ 0"
1550 P(5,6)="-826 0C+ 2+ 0- 4+ 0"
1560 P(5,7)="802 0S+ 0+ 0+ 2+ 0"
1570 P(5,8)="425 0S+ 1+ 0- 2+ 0"
1580 P(5,9)="-229 1C+ 0+ 0+ 1+ 0"
1590 P(5,10)="-153 0C+ 2+ 0- 6+ 0"
1600 P(5,11)="-142 1S+ 0+ 0+ 1+ 0"
1610 P(5,12)="-114 0C+ 0+ 0+ 1+ 0"
1620 P(5,13)="101 1S+ 2+ 0- 5+ 0"
1630 P(5,14)="-70 0C+ 0+ 2+ 0+ 0"
1640 P(5,15)="67 0S+ 0+ 2+ 0+ 0"
1650 P(5,16)="66 0S+ 2+ 0- 6+ 0"
1660 P(5,17)="60 1C+ 2+ 0- 5+ 0"
1670 P(5,18)="41 0S+ 1+ 0- 3+ 0"
1680 P(5,19)="39 0S+ 0+ 0+ 3+ 0"
1690 P(5,20)="31 0S+ 1+ 0- 1+ 0"
1700 P(5,21)="31 0S+ 2+ 0- 2+ 0"
1710 P(5,22)="-29 0C+ 2+ 0- 3+ 0"
1720 P(5,23)="-28 0S+ 2+ 0- 6+ 3"
1730 P(5,24)="28 0C+ 1+ 0- 3+ 0"
1740 P(5,25)="22 1S+ 2+ 0- 4+ 0"
1750 P(5,26)="-22 0S+ 0+ 0+ 1- 3"
1760 P(5,27)="20 0S+ 2+ 0- 3+ 0"
1770 P(5,28)="20 0C+ 4+ 0-10+ 0"
1780 P(5,29)="19 0C+ 0+ 0+ 2- 3"
1790 P(5,30)="19 0S+ 4+ 0-10+ 0"
1800 P(5,31)="-17 1C+ 0+ 0+ 2+ 0"
1810 P(5,32)="-16 0C+ 0+ 0+ 1- 3"
1820 P(5,33)="-12 0S+ 2+ 0- 4+ 0"
1830 P(5,34)="12 0C+ 1+ 0+ 0+ 0"
1840 P(5,35)="-12 0S+ 0+ 0+ 2- 2"
1850 P(5,36)="-11 1S+ 0+ 0+ 2+ 0"
1860 P(5,37)="-11 0C+ 2+ 0- 7+ 0"
1870 P(5,38)="10 0S+ 0+ 0+ 2- 3"
1880 P(5,39)="10 0C+ 2+ 0- 2+ 0"
1890 P(5,40)="9 0S+ 4+ 0- 9+ 0"
1900 P(5,41)="-8 0S+ 0+ 0+ 1- 2"
1910 P(5,42)="-8 0C+ 0+ 2+ 1+ 0"
1920 P(5,43)="8 0C+ 0+ 2- 1+ 0"
1930 P(5,44)="8 0C+ 0+ 0+ 1- 1"
1940 P(5,45)="-8 0S+ 0+ 2- 1+ 0"
1950 P(5,46)="7 0S+ 0+ 2+ 1+ 0"
1960 P(5,47)="-7 0C+ 1+ 0- 2+ 0"
1970 P(5,48)="-7 0C+ 0+ 0+ 2+ 0"
1980 P(5,49)="-6 1S+ 4+ 0-10+ 0"
1990 P(5,50)="6 1C+ 4+ 0-10+ 0"
2000 '

    Uranus
2010 P(6,0)="1922252628"
2020 P(6,1)="19397 0S+ 0+ 0+ 1+ 0+ 0"
2030 P(6,2)="570 0S+ 0+ 0+ 2+ 0+ 0"
```

More

Listing continued.

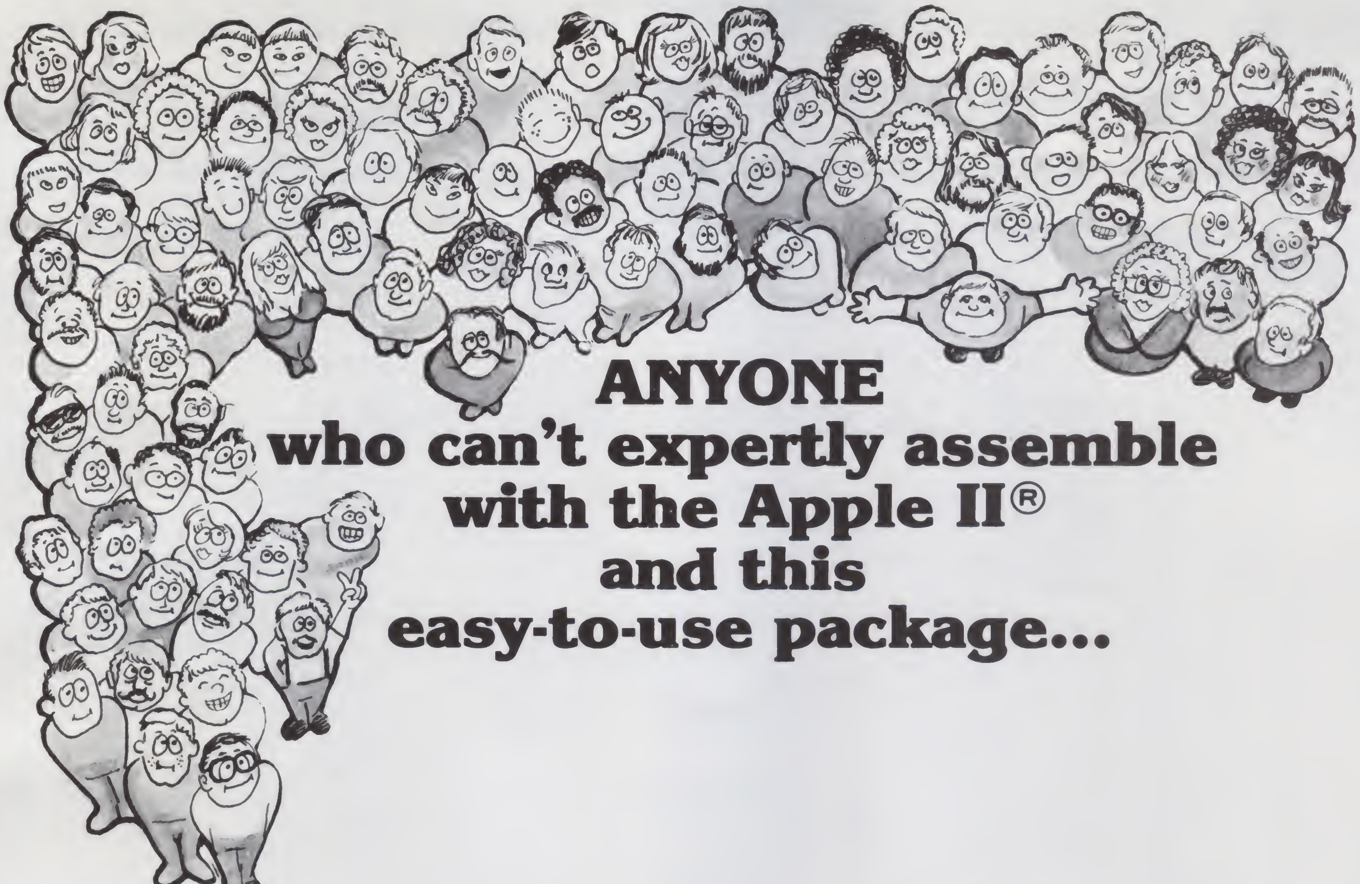
```
2040 P(6,3)="-536 1C+ 0+ 0+ 1+ 0+ 0"
2050 P(6,4)="143 0S+ 0+ 1- 2+ 0+ 0"
2060 P(6,5)="110 1S+ 0+ 0+ 1+ 0+ 0"
2070 P(6,6)="102 0S+ 0+ 1- 3+ 0+ 0"
2080 P(6,7)="76 0C+ 0+ 1- 3+ 0+ 0"
2090 P(6,8)="-49 0S+ 1+ 0- 1+ 0+ 0"
2100 P(6,9)="32 2C+ 0+ 0+ 0+ 0+ 0"
2110 P(6,10)="-30 1C+ 0+ 0+ 2+ 0+ 0"
2120 P(6,11)="29 0S+ 2- 6+ 3+ 0+ 0"
2130 P(6,12)="29 0C+ 0+ 0+ 2+ 0- 2"
2140 P(6,13)="-28 0C+ 0+ 0+ 1+ 0- 1"
2150 P(6,14)="23 0S+ 0+ 0+ 3+ 0+ 0"
2160 P(6,15)="-21 0C+ 1+ 0- 1+ 0+ 0"
2170 P(6,16)="20 0S+ 0+ 0+ 1+ 0- 1"
2180 P(6,17)="20 0C+ 0+ 1- 2+ 0+ 0"
2190 P(6,18)="-19 0C+ 0+ 1- 1+ 0+ 0"
2200 P(6,19)="17 0S+ 0+ 0+ 2+ 0- 3"
2210 P(6,20)="14 0S+ 0+ 0+ 3+ 0- 3"
2220 P(6,21)="13 0S+ 0+ 1- 1+ 0+ 0"
2230 P(6,22)="-12 2C+ 0+ 0+ 1+ 0+ 0"
2240 P(6,23)="-12 0C+ 0+ 0+ 1+ 0+ 0"
2250 P(6,24)="10 0S+ 0+ 0+ 2+ 0- 2"
2260 P(6,25)="-9 0S+ 0+ 0+ 0+ 2+ 0"
2270 P(6,26)="-9 2S+ 0+ 0+ 1+ 0+ 0"
2280 P(6,27)="9 0C+ 0+ 0+ 2+ 0- 3"
2290 P(6,28)="8 1C+ 0+ 1- 2+ 0+ 0"
2300 P(6,29)="7 1C+ 0+ 1- 3+ 0+ 0"
2310 P(6,30)="-7 1S+ 0+ 1- 3+ 0+ 0"
2320 P(6,31)="7 1S+ 0+ 0+ 2+ 0+ 0"
2330 P(6,32)="6 0S+ 2- 6+ 2+ 0+ 0"
2340 P(6,33)="6 0C+ 2- 6+ 2+ 0+ 0"
2350 P(6,34)="5 0S+ 0+ 1- 4+ 0+ 0"
2360 P(6,35)="-4 0S+ 0+ 0+ 3+ 0- 4"
2370 P(6,36)="4 0C+ 0+ 0+ 3+ 0- 3"
2380 P(6,37)="-3 0C+ 0+ 0+ 0+ 0+ 1"
2390 P(6,38)="-2 0S+ 0+ 0+ 0+ 0+ 1"
2400 '
    Neptune
2410 P(7,0)="1922252829"
2420 P(7,1)="3523 0S+ 0+ 0+ 0+ 1+ 0"
2430 P(7,2)="-50 0S+ 0+ 0+ 0+ 0+ 2"
2440 P(7,3)="-43 1C+ 0+ 0+ 0+ 1+ 0"
2450 P(7,4)="29 0S+ 1+ 0+ 0- 1+ 0"
2460 P(7,5)="19 0S+ 0+ 0+ 0+ 2+ 0"
2470 P(7,6)="-18 0C+ 1+ 0+ 0- 1+ 0"
2480 P(7,7)="13 0C+ 0+ 1+ 0- 1+ 0"
2490 P(7,8)="13 0S+ 0+ 1+ 0- 1+ 0"
2500 P(7,9)="-9 0S+ 0+ 0+ 2- 3+ 0"
2510 P(7,10)="9 0C+ 0+ 0+ 2- 2+ 0"
2520 P(7,11)="-5 0C+ 0+ 0+ 2- 3+ 0"
2530 P(7,12)="-4 1S+ 0+ 0+ 0+ 1+ 0"
2540 P(7,13)="4 0C+ 0+ 0+ 1- 2+ 0"
2550 P(7,14)="4 2S+ 0+ 0+ 0+ 1+ 0"
2560 '
    Pluto
2570 P(8,0)="3233"
2580 P(8,1)="101577 0S+ 1+ 0"
2590 P(8,2)="15517 0S+ 2+ 0"
2600 P(8,3)="-3593 0S+ 0+ 2"
2610 P(8,4)="3414 0S+ 3+ 0"
2620 P(8,5)="-2201 0S+ 1- 2"
2630 P(8,6)="-1871 0S+ 1+ 2"
2640 P(8,7)="839 0S+ 4+ 0"
2650 P(8,8)="-757 0S+ 2+ 2"
2660 P(8,9)="-285 0S+ 3+ 2"
2670 P(8,10)="227 2S+ 1+ 0"
2680 P(8,11)="218 0S+ 2- 2"
2690 P(8,12)="200 1S+ 1+ 0"
2700 '
    Problem Insertion
2710 PRINT"Insert Problem"
2720 PRINT"Planets are sequentially numbered."
2730 PRINT"   Mercury = 0."
2740 PRINT"   Venus   = 1."
2750 PRINT"   Pluto   = 8."
2760 PRINT"Number >8 = End.":PRINT""
2770 INPUT"Planet Number";A
2780 IF A>8 THEN END
2790 '
    Convert Gregorian calendar to Julian Date.
2800 INPUT"Month";IM
2810 INPUT"Day";ID
2820 INPUT"Year";IY
2830 IJD=367*IY
2840 IJD=IJD-FIX(7*(IY+FIX((IM+9)/12))/4)
2850 IJD=IJD-FIX(3*(FIX((IY+FIX((IM-9)/7))/100)+1)/4)
2860 IJD=IJD+FIX(275*IM/9)+ID+1721029#
2870 IJD=IJD-.5#'half day diff in standard
2880 ITT=IJD-2451545#'ITT=t
2890 IT=ITT/36525#+1#'IT=Time in Julian Centuries
2900 GOTO 2950
2910 'B(7-33)=Fundamental Arguments in dbl precision
```

More



Listing continued.

```
2920 B(V)=BA(V)+BB(V)*ITT
2930 B(V)=B(V)-FIX(B(V))'Get part of revolution
2940 RETURN
2950 '

    A=No. of planet being processed
2960 ON A+1 GOTO 2970,3000,3010,3020,3030,3040,3050,3060,3070
2970 V=9:V1=15:GOTO 3080'Mercury
2980 'V=No. of 'L' argument
2990 'V1=No. iterations for individual planet
3000 V=12:V1=6:GOTO 3080'Venus
3010 V=7:V1=12:GOTO 3080'Earth reciprocal of Sun
3020 V=15:V1=23:GOTO 3080'Mars
3030 V=18:V1=40:GOTO 3080'Jupiter
3040 V=21:V1=50:GOTO 3080'Saturn
3050 V=24:V1=38:GOTO 3080'Uranus
3060 V=27:V1=14:GOTO 3080'Neptune
3070 V=31:V1=12:GOTO 3080'Pluto
3080 GOSUB 2920
3090 BL(A)=B(V)*BP3'L in arc secs
3100 FOR C=1 TO LEN(P(A,0)) STEP 2
3110 '

    calculate Plon Trig Arguments
3120 V=VAL(MID$(P(A,0),C,2)):GOSUB 2920:B(V)=B(V)*BP2'b(v) in radians
3130 NEXT C
3140 BG(A)=BL(A)'bg(a)=lambda
3150 FOR V=1 TO V1'V1=No. of iterations
3160 V2=INSTR(P(A,V)," ")-1
3170 BCO=VAL(LEFT$(P(A,V),V2))'BCO=Coeficient peeled off data string
3180 ITF=VAL(MID$(P(A,V),V2+2,1))'ITF='T' factor peeled out of data string
3190 IF ITF<>0 THEN BCO=BCO*IT*ITF'Multiply by T factor if >0
3200 FOR V3=1 TO LEN(P(A,0))/2
3210 BV=VAL(MID$(P(A,0),2*V3-1,2))
3220 BT=BT+VAL(MID$(P(A,V),V2+1+3*V3,3))*B(BV)
3230 NEXT V3
3240 '

    Do either sine or cosine

3250 IF MID$(P(A,V),V2+3,1)="C" THEN 3270
3260 BG(A)=BG(A)+BCO*SIN(BT):GOTO 3280
3270 BG(A)=BG(A)+BCO*COS(BT)
3280 BT=0:NEXT V
3290 IF A=2 THEN BG(A)=BG(A)+648000#'
    recip of Sun for Earth

3300 IF BG(A)>BP3 THEN BG(A)=BG(A)-BP3
3310 IF BG(A)<0 THEN BG(A)=BG(A)+BP3:GOTO 3310
3320 PRINT"Lambda="BG(A)"Arc Sec"
3330 BM=FIX(BG(A)/60#)
3340 BS=BG(A)-BM*60#
3350 BD=FIX(BM/60#)
3360 BM=BM-BD*60#:IF BD=-1 AND BM=60 THEN BD=0:BM=0
3370 PRINT USING P2;BD;:PRINT"# ";
3380 PRINT USING P1;BM;:PRINT"' ";
3390 PRINT USING P;BS;:PRINT CHR$(34)
3400 PRINT"":PRINT"Insert New Problem"
3410 GOTO 2720
```



These BASIC routines make arrays easy to handle.

MAT Functions

By B. Antony Paturzo

Arrays come in handy when handling large amounts of data, especially data that can be easily tabulated. So, it's not surprising to find arrays used in a large variety of programs, from engineering to business.

What *is* surprising, and unfortunate, is that the BASICs available for a large number of micros do not have MAT functions, which are the BASIC array manipulators. Without these functions, the user must either operate on elements of an array one by one, or write his own MAT functions.

```
1000 REM MAT READ
1010 DIM A(RA,CA)
1020 FOR I=1 TO RA
1030 FOR J=1 TO CA
1040 READ A(I,J)
1050 NEXT J
1060 NEXT I
1070 RETURN
```

Listing 1.

```
1080 REM MAT INPUT
1090 DIM A(RA,CA)
1100 FOR I=1 TO RA
1110 PRINT "ROW ";I
1120 FOR J=1 TO CA
1130 PRINT "COLUMN ";J
1140 INPUT A(I,J)
1150 NEXT J
1160 NEXT I
1170 RETURN
```

Listing 2.

First, an array is a collection of items, which can be anything from profit percentages to resistor values. A one-dimensional array is called a *list* of items, or in mathematical terms, a *vector*. A two-dimensional array is called a *table*, or matrix. Since vectors and matrices both describe arrays, you can think of a vector as a special kind of matrix.

Keeping the above definitions in mind, the following example programs are intended for those using a BASIC without MAT functions.

FOR-NEXT Loops

Most of the MAT functions can be implemented using FOR-NEXT loops. Listing 1 shows how a MAT READ operation can be done. RA and CA refer to row and column, respectively, and are defined by the calling program. The DATA statements, contained elsewhere, can be written so that an RA number of DATA statements each contain a CA number of elements.

A variation of the read subroutine permits user input of the matrix elements. Listing 2 shows how you can input the matrix elements individually, a process obviously slower than using DATA statements.

A MAT PRINT is nearly identical to the MAT READ subroutine, with the replacement of READ with PRINT in line 1040. Rather than duplicate the read subroutine for MAT PRINT, Listing 3 shows how you can combine the two operations using W (for write) as the calling program selector of either a read or print operation.

The subroutine for a MAT ADD/-SUBTRACT is shown in Listing 4, with S (for subtract) being the calling program's selection of either an add or subtract operation. The result is stored in matrix D. Note that for these two operations, matrix A must contain the same number of rows as matrix B, as well as the same number of columns as B.

Matrix multiplication is not as straightforward as addition or subtraction. This type of multiplication consists of a *row-on-column* multiplication sequence. Therefore, matrix A's number of columns must equal matrix B's number of rows. The result of the multiplication, matrix D, will be a matrix with the same number of rows as matrix A and the same number of columns as matrix B. Two cases are illustrated in Listing 5, the first where B is a vector (or one-dimensional), and the second where B is two-dimensional.

The remaining MAT functions are MAT ZER, MAT CON, MAT IDN, MAT TRN and MAT INV, and the DET function, which finds the determinant of a *square* matrix. MAT ZER and MAT CON are relatively simple operations; the former assigns all 0s to a matrix, while the latter assigns all 1s to a matrix. MAT ZER can be accomplished by replacing line 1040 of Listing 1 with "A(I,J) = 0," while MAT CON replaces line 1040 with "A(I,J) = 1."

MAT IDN creates an identity ma-

Address correspondence to B. Antony Paturzo, 1929 Trudie Drive, San Pedro, CA 90732.

```
1180 REM MAT READ/PRINT
1190 REM IF W(WRITE)=1, PRINT
1200 DIM A(RA,CA)
1210 FOR I=1 TO RA
1220 FOR J=1 TO CA
1230 IF W=1 THEN PRINT A(I,J):GOTO 1250
1240 READ A(I,J)
1250 NEXT J
1260 NEXT I
1270 RETURN
```

Listing 3.

```
1380 REM MAT MULT, B IS A VECTOR
1390 REM SU IS SUM OF MULTIPLIED ELEMENTS
1400 DIM D(RB)
1410 SU=0
1420 FOR I=1 TO RA
1430 FOR J=1 TO CA
1440 SU=SU+(A(I,J)*B(J))
1450 NEXT J
1460 D(I)=SU
1470 SU=0
1480 NEXT I
1490 RETURN
```

Listing 5a.

```
1280 REM MAT ADD/SUBTRACT
1290 REM IF S(SUBTRACT)=1, SUBTRACT
1300 DIM D(RA,CA)
1310 FOR I=1 TO RA
1320 FOR J=1 TO CA
1330 IF S=1 THEN D(I,J)=A(I,J)-B(I,J):GOTO 1350
1340 D(I,J)=A(I,J)+B(I,J)
1350 NEXT J
1360 NEXT I
1370 RETURN
```

Listing 4.

```
1500 REM MAT MULT, B IS A MATRIX
1510 DIM D(RA,CB)
1520 SU=0
1530 FOR I=1 TO RA
1540 FOR J=1 TO CB
1550 FOR K=1 TO CA
1560 SU=SU+(A(I,K)*B(K,J))
1570 NEXT K
1580 D(I,J)=SU
1590 SU=0
1600 NEXT J
1610 NEXT I
1620 RETURN
```

Listing 5b.

trix; that is, a square matrix containing all 0s, except for the elements on its main diagonal, which are all 1s. A property of the identity matrix is that if a matrix is multiplied by an identity matrix of equal dimension, the result will be the original matrix. Thus, the identity matrix behaves in matrix multiplication much like the constant 1 does in ordinary multiplication.

The simplest way to get an identity matrix is to first call MAT ZER and then the subroutine shown in Listing 6. MAT ZER will fill the matrix with all 0s, and the subroutine in Listing 6 will put 1s along the main diagonal of the matrix.

```
1630 REM PUT 1S IN MAIN DIAGONAL
1640 FOR I=1 TO RA
1650 J=I
1660 A(I,J)=1
1670 NEXT I
1680 RETURN
```

Listing 6.

Listing 7 shows the subroutine for MAT TRN, which causes the rows and columns of a matrix to be transposed or interchanged. Matrix B is used to hold the transposed matrix; matrix A, the matrix to be transposed, is left unaltered, which lets you use both the original and the transposed matrices in any subsequent operations.

```
1690 REM MAT TRN
1700 DIM B(CA,RA)
1710 FOR I=1 TO RA
1720 FOR J=1 TO CA
1730 B(J,I)=A(I,J)
1740 NEXT J
1750 NEXT I
1760 RETURN
```

Listing 7.

MAT INV finds the inverse of a square matrix. The product of a matrix and its inverse gives an identity matrix. The subroutine shown in Listing 8 uses *inversion-in-place,* meaning that the original matrix is replaced with its inverted form as the inversion process takes place. This method uses less memory than methods that require a temporary storage matrix. One drawback, however, is that the original matrix is lost; therefore, if the original matrix is needed for further operations, it should be saved in another matrix prior to calling MAT INV.

The subroutine for the final MAT function, the one used to find the determinant of a square matrix, is shown in Listing 9. Once again, matrix A is altered during this operation, so if you need the original matrix A for other operations, save it prior to calling the DET function. The numerical value for the determinant is returned in the variable S.

An Example

Now that you have all of the MAT functions available, a simple example will illustrate their usefulness. Suppose that you have the following set of equations:

$2X_1 - 2X_2 + 5X_3 = 13$
$2X_1 + 3X_2 + 4X_3 = 20$
$3X_1 - 1X_2 + 3X_3 = 10$

You would like to determine the values for X_1, X_2 and X_3 such that all of the equations are satisfied. If you fill matrix A with the coefficients of X_1, X_2 and X_3 so that:

$$A = \begin{matrix} 2 & -2 & 5 \\ 2 & 3 & 4 \\ 3 & -1 & 3 \end{matrix}$$

and if you fill matrix B with the right hand sides of the three equations so that:

$$B = \begin{matrix} 13 \\ 20 \\ 10 \end{matrix}$$

you can determine the values for X_1, X_2 and X_3 by the equation $D = INV\ A \times B$. In other words, matrix D, which contains the numerical values for X_1, X_2 and X_3, can be found by taking the inverse of the matrix containing the coefficients (matrix A) and then multiplying this inverse by matrix B, which contains the right hand sides of the equations.

A program to do all of this is shown in Listing 10. Line 2130 contains a subroutine that is simply a routine to MAT READ vector B (see Listing 11).

This example could easily have been done by hand using algebra, but imagine having to solve ten simultaneous equations, or even 20, by hand. At that point, you'll really appreciate your MAT functions.■

```
1770 REM MAT INV
1780 REM MAT A IS REPLACED WITH ITS INVERSE
1790 FOR K=1 TO RA
1800 FOR J=1 TO RA
1810 IF J=K GOTO 1830
1820 A(K,J)=A(K,J)/A(K,K)
1830 NEXT J
1840 A(K,K)=1/A(K,K)
1850 FOR I=1 TO RA
1860 IF I=K GOTO 1910
1870 FOR J=1 TO RA
1880 IF J=K GOTO 1900
1890 A(I,J)=A(I,J)-A(K,J)*A(I,K)
1900 NEXT J
1910 NEXT I
1920 FOR I=1 TO RA
1930 IF I=K GOTO 1950
1940 A(I,K)=-A(I,K)*A(K,K)
1950 NEXT I
1960 NEXT K
1970 RETURN
```

Listing 8.

```
1980 REM DET FUNCTION
1990 FOR I=1 TO RA-1
2000 FOR J=I+1 TO RA
2010 Q=A(J,I)/A(I,I)
2020 FOR K=1 TO RA
2030 A(J,K)=A(J,K)-(A(I,K)*Q)
2040 NEXT K
2050 NEXT J
2060 NEXT I
2070 S=1
2080 FOR I=1 TO RA
2090 S=S*A(I,I)
2100 NEXT I
2110 REM S IS VALUE FOR DETERMINANT
2120 RETURN
```

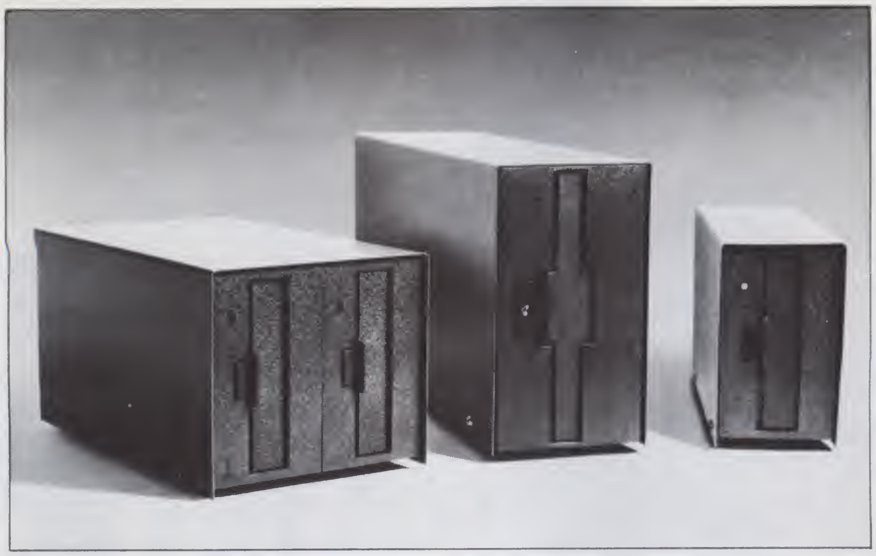
Listing 9.

```
10 REM SOLVE 3 SIMULTANEOUS EQUATIONS
15 REM READ MATRIX A
20 RA=3
30 CA=3
40 DATA 2,-2,5
50 DATA 2,3,4
60 DATA 3,-1,3
70 GOSUB 1000
75 REM READ VECTOR B
80 RB=3
90 DATA 13,20,10
100 GOSUB 2130
105 REM FIND INVERSE OF MATRIX A
110 GOSUB 1770
115 REM MULTIPLY INVERSE A AND VECTOR B
120 X=RB
125 PRINT
130 GOSUB 1380
135 REM OUTPUT RESULT
140 FOR I=1 TO RA
150 PRINT "X";I;"=";D(I)
160 NEXT I
170 END

OK
run

X 1 = 1
X 2 = 2
X 3 = 3
```

Listing 11.

```
2130 REM READ VECTOR B
2140 DIM B(RB)
2150 FOR I=1 TO RB
2160 READ B(I)
2170 NEXT I
2180 RETURN
```

Listing 10.



More power to you.

Disk drives, hardware and software— now more affordable *and* more available!

NEW PRODUCT ANNOUNCEMENT: NEW MODEL III WITH DISK DRIVES!

TRS-80™ Model III with dual MPI B91 80-track disk drives—704K of reliable disk storage on only two drives!
This system features:
- Two MPI B91 double-density 80 track drives
- 48K RAM, double-density Newdos 80
- 3 millisecond access time
- Includes A.M. Electronics' controller board and MAKE80© program
- Complete, ready-to-run

Special Introductory Price **$2295**

The Ultimate Small Business Computer: TRS-80™ Model III with four dual-headed MPI B92 80-track disk drives!
This system features:
- Four proven MPI drives
- 48K RAM, double-density Newdos 80
- 3 millisecond access time
- 2.8 Megabytes reliable disk storage
- Includes A.M. Electronics' MAKE80© program and controller board
- Complete, ready-to-run

Special Introductory Price **$3795**

CASES AND POWER SUPPLIES

5½-inch enclosures
Single drive unit case and power supply **$85**
Dual drive unit case and power supply **$120**
(Extender cables are $15 each extra)

8-inch enclosures
Single drive unit case and power supply **$150**

Store Hours: Tues. - Fri. 11-7, Sat. 11-5.
Complete Service Center: For speedy repairs, including modified TRS-80's.

Attention dealers, OEM's & distributors
Call us for details on our attractive pricing.

SPECIALS!

SAVE ON COMPLETE DISK DRIVES!

5¼-inch drives
40-track MPI 51 w/case, power supply and extender cable ... **$320**
80-track MPI 91 w/case, power supply and extender cable ... **$425**
Dual-headed 80-track drive with case and power supply **$550**

8-inch drives
Single Siemens 8-inch drive with dual case and power supply .. **$695**

INTRODUCING TEAC DISK DRIVES—THE BEST DRIVES WE HAVE SEEN TO DATE!

Teac 40-track disk drive **$299**
Teac 80-track disk drive **$399**
All drives complete with cases, power supplies and extender cables.
(Add $20 for "Flippy" version)

PRINTER SPECIAL!
Epson MX-80 Printer **$495**

TRS-80™ SOFTWARE PACKAGES

MAKE80©
Converts standard 35 or 40 track diskettes into 80-track-readable diskettes ... **$14.95**

The power behind the drives©

✓39

A.M. ELECTRONICS, INC.
3366 Washtenaw Ave.
Ann Arbor, Michigan 48104
(313) 973-2312

Visit our retail showroom for a "hands-on" look at our TRS-80™ hardware and software

SUPER UTILITY, by Kim Watt.
Stand-alone 24K machine language program for disk includes:

ZAP UTILITY
- Read/modify data regardless of disk protection • One-step track-to-track/sector to sector • Dual cursors; ASCII and Hex. Modify in Hex, Decimal or ASCII • Display disk sectors, display file sectors, copy disk sectors, compare disk sectors, display/modify main memory, search memory or disk for specified string and return location

PURGE UTILITY
- Kill files by file spec or category • Zero out unused directories or sectors
- Compute passwords, change disk's name, date, passwords, protection levels • Directory routine indicates all active and inactive files, their location in directory and status of granules

DISK COPY UTILITY
- Copy *any* TRS-80™ readable disk, regardless of protection

TAPE COPY UTILITY
- Copy *any* TRS-80™ readable tape, regardless of protection or baud rate

DISK REPAIR UTILITY
- *Automatically* repair damaged HIT, GAT or BOOT sectors • Directory check advises of errors • Automatic recovery of killed files • Shows active and inactive files, and their location on the disk

MEMORY UTILITY
- Move memory • Jump to memory • Test memory • Compare memory • Input or output any byte to any port • Zero memory • Exchange memory • Edit memory • Load memory

SUPER UTILITY is now available for **$49.95,** plus $2.50 shipping and handling!

™TRS-80 is a trademark of Tandy Corp.

VISA

master charge

✓ Reader Service—see page 178

Abbreviated keywords allow more efficient programming for owners of early PETs.

PET Shorthand Compleat

By Gary L. Ratliff

The idea of shorthand is nothing new to old-time computer users. My first exposure to computer programming 15 years ago was through a book that introduced a 24-bit code. All the coding was in binary, which was tedious to use. Hexadecimal and octal are used as shorthand for the more cumbersome binary. Assembly-language programming is still a further approach that makes the operation codes of the computer more meaningful for human users.

Shorthand is an integral part of some high-level languages and an added feature of several of the BASIC-speaking computers for the home user, "personal computer" market. APL has a number of its functions available with a single keystroke. FORTH offers the user the ability to virtually define his program using as short a "word" as he desires. The Bally Arcade offers its BASIC with all keywords available at a single stroke of one key. The shorthand of the Level I TRS-80 is useful and well documented in the user's manual. The shorthand is no longer available once the owner upgrades to the more versatile Level II system.

That the Commodore PET also has available a shorthand that may be used either within a program or in the direct mode is not so well known. Its presence was mentioned in an early issue of *Cursor*, the cassette magazine for PET users. The purpose of this article is to expand on that original idea and explain its use and its limitations.

Gary L. Ratliff, 101 East St., Mendenhall, MS 39114.

PET's Secret Code

The idea is very simple. Most keywords in PET BASIC may be obtained by entering the first letter of the word followed by the shifted second letter of the word. Since the shifted graphic characters are difficult to reproduce, the tables and the examples presented assume that the user has set his computer in the lowercase alphabetic mode. This is done by entering the direct statement: POKE% ($¢*,!$. This is exactly what the screen of your PET may look like using shorthand if you fail to change modes. By changing modes this becomes readable: POKE 59468, 14. Now the second letter of the keyword shifted appears to be a lowercase letter.

Just a little thought and some time spent in constructing an alphabetic list of the PET keywords will clearly reveal why this method won't work for all keywords. The most obvious fact is that many of the keywords are only two bytes anyway. Therefore, even if this method works for them, there is nothing to gain by using this method on words only two letters long.

The next exception is obviously keywords that begin with the same first two letters, for example, STOP, STEP, STR$ or INPUT, INPUT#, INT. Many of these exceptions may still be successfully converted into shorthand by differentiating them on the third letter. St is shorthand for STOP; while the three letter code STe gives STEP and STr gives STR$. Table 1 shows a list of PET BASIC keywords and their shortened versions.

Now that we are equipped with a table of values to convert programs into this shorthand, let's test our new knowledge by trying a short example:

```
10 OPEN 1,1,1,"DATA"
20 FOR I=1 TO 25
30 PRINT#1,I
40 PRINT I
50 NEXT
60 CLOSE1
70 PRINT "DONE";TAB(15)"STOP"
80 END
```

Using the multiline feature of the Microsoft BASICs and the PET shorthand feature, this entire program may be entered as a one-liner as follows:

```
50p1,1,1,"DATA":FoI=1TO25:Pr1,I:?I:Ne:CL
o1:?"DONE";Ta15)"STOP":En
```

Before running this program, remove your cassette tape from your tape drive. The program creates a file called DATA which contains the first positive integers from 1 to 25. If you have a tape containing several programs and you run this program, it will create this file on top of your program tape. If this was a commercial

program for which you paid several dollars, you would not be happy to have the program destroy one of your tapes.

Now type Ru and hit return. You will read the numbers on the screen and hear the cassette motor turn on and then off. Now type Li and hit return. Notice that the shorthand you entered has been expanded to the full PET keywords.

One of the principle uses for this shorthand is the effective lengthening of the BASIC input buffer. This buffer is 80 bytes or two screen lines. If the full text of the keywords is typed, then maximum use of a line cannot be realized. By using shorthand the fact that user-defined functions may only be one line in length may be reduced. Therefore, with shorthand, more involved user functions may be created.

To demonstrate how the buffer is lengthened, let us consider another example. The following is a one-line direct statement that will give an ASCII dump of the variables contained within a BASIC program:

```
FORI=256*PEEK(125)+PEEK(124)TO256*P
EEK(127)+PEEK(126)STEP7:PRINTPEEK(I);P
EEK(I+1):NEXT
```

This statement is 86 bytes. Even using the ? for PRINT will only reduce it to 82 bytes. But with PET shorthand it is only 66 bytes. Now the statements will work as a direct statement and the extra luxury of 14 bytes even allows us to include a timing loop to slow down the listing:

```
FoI=256*Pe(125)+Pe(124)TO256*Pe(127)+P
e(126)STe7:?Pe(I);Pe(I+1):FoJ=1TO999:Ne:Ne
```

By the use of shorthand in this example, we have created a direct statement 103 bytes in length. This means that we have effectively increased the input buffer by 28.75 percent.

One additional technique that may prove of value to persons who want to design complex functions through the use of the user-defined function feature is spoon-feeding the line into the program. The user-defined function should be the first line entered. The line is translated into its token symbols and its numerical ASCII values. (Jim Butterfield's tables (see References) make this task much easier.)

Next, these token values are directly poked into the program via direct statements. Be sure to follow the end of the line with an additional poking of two zeros. This is to signal the BASIC interpreter that there are no further lines to follow. Failure to do this step will result in a system crash in most cases. Now list, and you should see this line as the first and only line of the program text. The rest of the program may be entered using the more conventional keyboard entry of the text.

Key	Short	Key	Short	Key	Short
ABS	Ab	INPUT#	In	READ	Re
AND	An	INPUT	N/A	RESTORE	REs
ASC	As	INT	N/A	RETURN	REt
ATN	At			RIGHT$	Ri
		LEN	N/A	RND	Rn
CHR$	Ch	LET	Le	RUN	Ru
CLOSE	CLo	LEFT$	LEf		
CLR	Cl	LIST	Li	SAVE	Sa
CMD	Cm	LOAD	Lo	SGN	Sg
CONT	Co	LOG	N/A	SIN	Si
COS	N/A			SPC	Sp
		MID$	Mi	SQR	Sq
DATA	Da			STOP	St
DEF	De	NEXT	Ne	STEP	STe
DIM	Di	NEW	N/A	STR$	STr
		NOT	No		
END	En			TAB	Ta
EXP	Ex	ON	N/A	TAN	N/A
		OR	N/A	THEN	Th
FOR	Fo	OPEN	Op	TO	N/A
FRE	Fr				
		PEEK	Pe	USR	Us
GET	Ge	POKE	Po		
GET#	N/A	POS	N/A	VAL	Va
GOTO	Go	PRINT#	Pr	VERIFY	Ve
GOSUB	GOs	PRINT	?		
				WAIT	Wa
IF	N/A	REM	N/A		

Notes: Both TAB and SPC have the leftmost parentheses included when this shorthand is used. Therefore, PRINT TAB (15);"ACCT.NO." should be coded using this method: ?Ta15);"ACCT.NO."

CONT is only usable as a command, and, if used within a program, it gives an error condition which must be removed.

Table 1. PET BASIC keywords and their shortened versions.

To demonstrate this method without giving a specific example, it is suggested that if the reader has access to the Mike Stone article on program overlays, which was reprinted in the *PET Gazette*, he try using the direct POKE technique with the program listed in the article. The linking and translation of the token symbols and ASCII values is already done. The correct start for this example is:

```
POKE1025,11:POKE1026,4:POKE1027,10:PO
KE1028,0:POKE1028,65 etc.
```

Advantages

1. There is a lesser chance of error as less material must be typed. If you *examine* all your key-in errors you will probably find that most of them occur closer toward the end of the word.

2. Expands the range of the line by allowing more lines to be used with the multiline feature already in this machine.

3. Shorter keywords should mean that once the technique is learned it should be possible to enter your programs faster.

4. More complex user-defined functions and direct statements. (The usefulness of direct statements in debugging may be enhanced by including debugging routines within the text of the program. Then these tests may be performed from the keyboard by a direct command GOsX, where X is the line number of the debugging routine. However, none of these subroutines should call on the INPUT function.)

Disadvantage

The main weakness of using PET shorthand is that lines that expand to more than 80 bytes may not be edited with the screen editor. The longest possible line with PET BASIC will list out at 242 bytes. This line would consist of a line number and ?:. A one-digit line number would allow the

OSI **SOFTWARE FOR OSI** OSI

A JOURNAL FOR OSI USERS!!

The Aardvark Journal is a bimonthly tutorial for OSI users. It features programs customized for OSI and has run articles like these:

1) Using String Variables.

2) High Speed Basic On An OSI.

3) Hooking a Cheap Printer To An OSI.

4) An OSI Disk Primer.

5) A Word Processor For Disk Or Tape Machines.

6) Moving The Disk Directory Off Track 12.

Four back issues already available!
$9.00 per year (6 issues)

ADVENTURES

Adventures are interactive fantasies where you give the computer plain English commands (i.e. take the sword, look at the control panel.) as you explore alien cities, space ships, ancient pyramids and sunken subs. Average playing time is 30 to 40 hours in several sessions. There is literally nothing else like them — except being there yourself. We have six adventures available.

ESCAPE FROM MARS — Explore an ancient Martian city while you prepare for your escape.

NUCLEAR SUBMARINE — Fast moving excitement at the bottom of the sea.

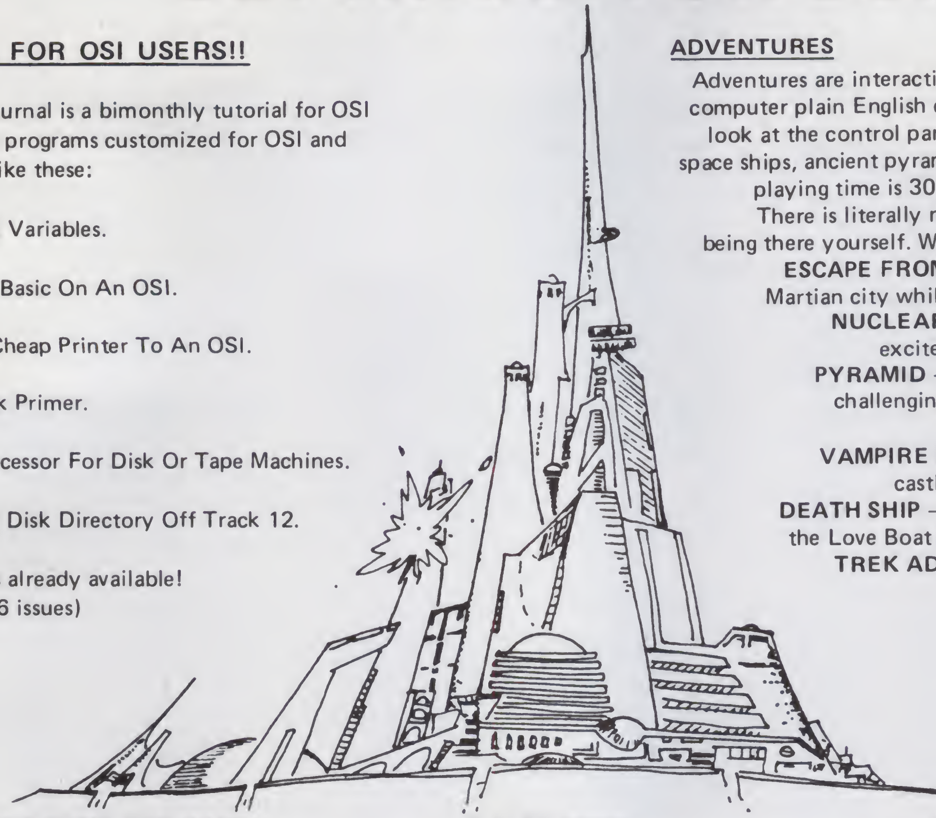
PYRAMID — Our most advanced and most challenging adventure. Takes place in our own special ancient pyramid.

VAMPIRE CASTLE — A day in old Drac's castle. But it's getting dark outside.

DEATH SHIP — It's a cruise ship — but it ain't the Love Boat and survival is far from certain.

TREK ADVENTURE — Takes place on a familiar starship. Almost as good as being there.

$14.95 each



NEW SUPPORT ROMS FOR BASIC IN ROM MACHINES

C1S — for the C1P only, this ROM adds full screen edit functions (insert, delete, change characters in a basic line.), Software selectable scroll windows, two instant screen clears (scroll window only and full screen.), software choice of OSI or standard keyboard format, Bell support, 600 Baud cassette support, and a few other features. It plugs in in place of the OSI ROM. NOTE: this ROM also supports video conversions for 24, 32, 48, or 64 characters per line. All that and it sells for a measly $39.95.

C1E/C2E for C1/C2/C4/C8 Basic in ROM machines.
This ROM adds full screen editing, software selectable scroll windows, keyboard correction (software selectable), and contains an extended machine code monitor. It has breakpoint utilities, machine code load and save, block memory move and hex dump utilities. A must for the machine code programmer replaces OSI support ROM. Specify system $59.95

DISK UTILITIES

SUPER COPY — Single Disk Copier
This copy program makes multiple copies, copies track zero, and copies all the tracks that your memory can hold at one time — up to 12 tracks at a pass. It's almost as fast as dual disk copying. — $15.95

MAXIPROSS (WORD PROCESSOR) — 65D polled keyboard only - has global and line edit, right and left margin justification, imbedded margin commands, choice of single, double or triple spacing, file access capabilities and all the features of a major word processor — and it's only $39.95.

P.C. BOARDS

MEMORY BOARDS!! — for the C1P. — and they contain parallel ports!
Aardvarks new memory board supports 8K of 2114's and has provision for a PIA to give a parallel ports! It sells as a bare board for $29.95. When assembled, the board plugs into the expansion connector on the 600 board. Available now!

PROM BURNER FOR THE C1P — Burns single supply 2716's. Bare board — $24.95.

MOTHER BOARD — Expand your expansion connector from one to five connectors or use it to adapt our C1P boards to your C4/8P. - $14.95.

ARCADE AND VIDEO GAMES

ALIEN INVADERS with machine code moves — for fast action. This is our best invaders yet. The disk version is so fast that we had to add selectable speeds to make it playable.
Tape - $10.95 — Disk - $12.95

TIME TREK (8K) — real time Startrek action. See your torpedoes move across the screen! Real graphics — no more scrolling displays. $9.95

STARFIGHTER — a real time space war where you face cruisers, battleships and fighters using a variety of weapons. Your screen contains working instrumentation and a real time display of the alien ships. $6.95 in black and white - $7.95 in color and sound.

MINOS — A game with amazing 3D graphics. You see a maze from the top, the screen blanks, and then you are in the maze at ground level, finding your way through on foot. Realistic enough to cause claustrophobia. — $12.95

SCREEN EDITORS

These programs all allow the editing of basic lines. All assume that you are using the standard OSI video display and polled keyboard.

C1P CURSOR CONTROL — A program that uses no RAM normally available to the system. (We hid it in unused space on page 2). It provides real backspace, insert, delete and replace functions and an optional instant screen clear.
$11.95

C2/4 CURSOR. This one uses 366 BYTES of RAM to provide a full screen editor. Edit and change lines on any part of the screen. (Basic in ROM systems only.)

FOR DISK SYSTEMS — (65D, polled keyboard and standard video only.)

SUPERDISK. Contains a basic text editor with functions similar to the above programs and also contains a renumberer, variable table maker, search and new BEXEC* programs. The BEXEC* provides a directory, create, delete, and change utilities on one track and is worth having by itself. — $24.95 on 5" disk - $26.95 on 8".

AARDVARK IS NOW AN OSI DEALER!

Now you can buy from people who can support your machine.

—THIS MONTH'S SPECIALS—

Superboard II	$279
C1P Model II	429
C4P	749

. . . and we'll include a free Text Editor Tape with each machine!

Video Modification Plans and P.C. Boards for C1P as low as $4.95

VISA

This is only a partial listing of what we have to offer. We now offer over 100 programs, data sheets, ROMS, and boards for OSI systems. Our $1.00 catalog lists it all and contains free program listings and programming hints to boot.

master charge

OSI **Aardvark Technical Services • 1690 Bolton • Walled Lake, MI 48088** OSI
91
(313) 669-3110

Reader Service—see page 178

CALENDAR

Computer Contest

A national computer problem-solving contest, open to junior-high and high-school students, will be held May 2, 1981. Schools and organizations interested in participating in this contest, which is sponsored by the University of Wisconsin—Parkside, should contact Dr. Donald T. Piele, Associate Professor of Mathematics, University of Wisconsin—Parkside, Kenosha, WI 53141, by April 4.

Executive Computer Conference

The Executive Computer Conference, sponsored by *Infosystems*, will be held April 13–14, 1981, in Washington, D.C. It will focus on key aspects of the computer's contribution to organizational productivity from a senior management perspective. For further information, contact the conference chairman, Kendall Burroughs, The Executive Computer Conference, 1730 North Lynn St., Suite 400, Arlington, VA 22209, 703/521-6209.

Micros and Education

Microcomputers and Education Second Annual Tri-state Conference, sponsored by the Center for the Advancement of Teaching and Learning (CATALYST), will be held at the Jersey City State College on March 6 and 7, 1981. The conference, which will feature speakers, workshops and exhibits, will examine the uses of microcomputers in the school and college curriculum and ways to improve computer literacy at all levels of education. For more information, contact CATALYST, Microcomputer Conference, Jersey City State College, 2039 Kennedy Boulevard, Jersey City, NJ 07305, 201/547-3094 or 3098.

College Symposium

The 14th Annual Small College Computing Symposium will take place March 13–14, 1981, at Augustana College. For further information, contact Joseph Pagone, director of Computer Science Education, Augustana College, Sioux Falls, SD 57197.

use of 40 ?s and 39 :s. To list this on the screen will take just over six lines of the screen.

One method to avoid this difficulty is to start entering a line mid screen. Now when listing the line your shorthand version will not scroll off the top of the screen. If you now find an error, it will be possible to correct it in the shorthand version of the text on the screen.■

References

Dr. David A. Lien: *Users Manual Level I,* 1977, Radio Shack, Fort Worth, TX.

Leonard Gilman and Allen J. Rose, *APL, An Interactive Approach*, 1974, John Wiley and Sons, New York.

Notes to *Cursor* No. 2, August 1978, Ron Jeffries Publisher, Goleta, CA.

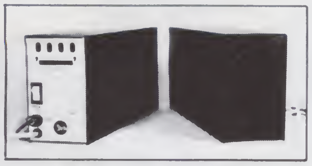
Mike Stone, "Program Overlays," *PET Gazette*, Vol. 1, No. 6.

PET User Manual, 1978, Commodore Business Machines, Palo Alto CA.

Jim Butterfield, *PET Gazette*, Spring 1979.

Gary Ratliff, "Just Poking Around With My PET," *Dr. Dobb's Journal PET User Manual,* 1978, Commodore Business Machines, Palo Alto, CA.

SIRIUS 8000 and 80 +
HIGH PERFORMANCE, LOW COST FLOPPY ADD-ONS!



The SIRIUS SYSTEMS 8000 and 80 + Series of Floppy Disk Add-Ons are designed to provide unmatched versatility and performance for your TRS-80* MOD I and MOD II.

80 + FOR YOUR MOD I—
COMMON CHARACTERISTICS include ■SINGLE/DOUBLE Density Operation ■90 Days P&L WARRANTY ■Mix any or all 80 + on the SS Standard Cable ■Exceptional Speed Stability ■5ms track-to-track access time (25ms for the 80 + 5) ■Switch Selectable Drive Address

The **SIRIUS 80 + 5** Is a SINGLE SIDED, 40 TRACK Disk Drive with 102K/204K Bytes Single/Double Density Operation. **SPECIAL INTRODUCTORY PRICE!!!**

SIRIUS 80 + 5............................**$299.95**
SIRIUS 80 + 5A
(with Floppy-Disk Option)............**314.95**

SIRIUS 80 + 1—a SINGLE SIDED, Fast 40 TRACK Disk Drive with 102/204K Bytes Single/Double Density Operation.

SIRIUS 80 + 1..............................**349.95**

SIRIUS 80 + 2—a DOUBLE SIDED, 80 TRACK (40/side) Disk Drive. It appears to the TRS-80* as TWO 40 TRACK drives yet COST LESS THAN HALF THE PRICE! This unit may require the SS Standard Cable. Formatted data storage is 204K/408K Bytes Single/Double Density.

SIRIUS 80 + 2..............................**429.95**

SIRIUS 80 + 3—a SINGLE SIDED, 80 Track (96 tpi) Disk Drive. Offering 2 1/3 times the storage of a standard Radio Shack Drive (204/408K Bytes Single/Double Density), the 80 + 3 reduces the need for diskettes tremendously!!

SIRIUS 80 + 3..............................**479.95**

SIRIUS 80 + 4—a DOUBLE SIDED, 160 track (80/side-96 tpi) Disk Drive. The ultimate in State-Of-The-Art 5 1/4" Floppy Disk Technology, the 80 + 4 is seen by the TRS-80* as TWO single sided drives. Thus, in terms of capacity, one 80 + 4 is equivalent to 4 2/3 standard Radio Shack drives—at a savings of over 73%. (With a Double Density Converter, the available memory is huge—408K/916K Bytes Single/Double Density!!)

SIRIUS 80 + 4..............................**629.95**

Note: The Expansion Interface limits drives to 12ms track-to-track

*TRS-80 of Tandy Corp.

ACCESSORIES
SS Standard 2 Drive Cable...........$29.95
5 1/4" Diskettes (Box of 10)............29.95
NEWDOS/80-Sophisticated Operating System for the TRS-80* from Apparat.......................................139.95

Save up to 10% *with these SIRIUS Packages!*

SIRIUS 80 + 5, NEWDOS/80, and Two Drive Cable..............$429.95
SIRIUS 80 + 3, NEWDOS/80, and Two Drive Cable..............**624.95**
SIRIUS 80 + 4, NEWDOS/80, and Two Drive Cable..............**749.95**
TWO SIRIUS 80 + 5's (SPECIAL INTRODUCTORY PRICE!!)..........**2 for/559.95**
TWO SIRIUS 80 + 3's, NEWDOS/80, and Two Drive Cable..........**999.95**
TWO SIRIUS 80 + 4's, NEWDOS/80, and Two Drive Cable..........**1349.95**
FOR YOUR MOD II—SIRIUS 8000 (NEW!!)
(All SIRIUS 8000 Series 8" Floppy Disk Drives include Case and Power Supply.)
SIRIUS 8000—SINGLE SIDED, 77 TRACK 8" Disk Drive..............**$649.95**
SIRIUS 8500—DOUBLE SIDED, 154 TRACK 8" Disk Drive..............**829.95**
SIRIUS 8100—2 DRIVE, SINGLE SIDED Expansion System..............**995.95**
SIRIUS 8150—2 DRIVE, DOUBLE SIDED Expansion System..............**1360.95**
2 Drive, 50 Conductor Cable for MOD II..............**34.95**

Introducing the Versatile, Low-Cost OMEGA Series Controller

As new technological advances bring down the cost of fast, reliable mass data storage, the need for an inexpensive, versatile controller have become greater and greater. To meet this need, SIRIUS SYSTEMS' OMEGA Series Controller was designed.

The SIRIUS OMEGA Series Controller Module utilizes an **on-board microprocessor** to mediate data transfer to a wide variety of peripherals from an equally wide variety of host computer systems. Up to four Winchester Hard Disks (8" or 14"), four 5¼" Floppy Disks Drives and/or up to eight 8" Floppy Disk Drives may be **in use at one time.**

SPECIFIC HARDWARE FEATURES INCLUDE:

- ■ *Control of up to twelve Floppy Disk Drives (eight 8" and/or four 5¼")*
  - *8" and/or 5¼" Disk Drive Utilization*
  - *Single (FM) or Double (MFM) density data storage*
  - *Hard or Soft sectored diskette usage*
  - *Utilization of "Quad" density (96 tpi) 8" or 5¼" Disk Drives*
- ■ *Control of up to four WINCHESTER type PRIAM DISKOS Disk Drives*
  - *8" or 14" may intermix on the same cable*
  - *Accommodates 8" and/or 14" drives of 10.6 Mbytes to 154 Mbytes*
  - *Ultra-Fast data transfers*
- ■ *Extremely flexible host-controller interfacing*

SPECIFIC SOFTWARE FEATURES INCLUDE:

- ■ Dynamic format modifications via command words
- ■ Extremely flexible format acceptance for unusual data storage formats
- ■ Easily interfaces to standard operating systems (TRS-DOS*, CP/M*, etc)
- ■ Operates in either get/put sector mode or data string mode
- ■ Performance parameters may be changed by EPROM replacement or Dynaminic Reprogramming

CP/M™ of Digital Research

THE OMEGA-S100—a DMA, Single Board controller available in Hard Disk, Floppy Disk and Combinational Versions.
OMEGA-S100 FDC (Floppy Disk only version)..............**$495.95**
OMEGA-S100 HDC (Hard Disk only version)..............**725.95**
OMEGA-S100 FHDC (Floppy and Hard Disk version)..............**975.95**
THE TRS-80* MOD II OMEGA—a Single Board, DMA Hard Disk Controller for the Radio Shack MOD II. OMEGA-MOD II..............**$725.95**
THE LSI-II OMEGA—a DMA, Single Board (quad width), DEC compatible Hard Disk controller for Q-BUS type computers (LSI-11/2 and LSI-11/23 compatible).
OMEGA-LSI 11..............**$1395.00**
NOTE: OMEGA Controllers available soon for the TRS-80* MOD I, STD BUS, and other systems. Consult SIRIUS SYSTEMS for current prices and availabilities.

PACKAGES (NEW!!!)

20 MegaByte, S-100 or MOD II system with Hard Disk Drive, Case & Power Supply, Cables, OMEGA Controller and Documentation—..............**$3615.**
33 MegaByte, S-100 or MOD II System (as above)..............**$4115.**
154 MegaByte, S-100 or MOD II System (as above)..............**$5820.**
CP/M™ 2.2 Operating System for above packages (specify S-100 or MOD II)........**$170.**

SPECIAL PURCHASE!!
NEW SIEMENS 8" and 5¼" Disk Drives at SUPER DISCOUNTS!

SIEMENS FDD-100-8 Single sided, 8" Single or Double Density Disk Drive..............**$389.95**
(2/$359.95 ea)
SIEMENS FDD-100-5 Single sided, 5¼" Single or Double Density Disk Drive..............**$249.95**
(2/$229.95 ea)
SIEMENS Technical Manual (specify 8" or 5¼")..............**$12.95**

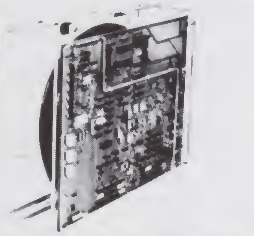
TFORTH! -what it has to offer YOU!

TFORTH is a procedural FORTH type language which specifies a process rather than a desired result. Designed to run on the TRS-80®, TFORTH is a very powerful tool by itself or used in conjunction with Assembly Programming. A rich set of WORDS come with TFORTH and many features considered as "extra with other FORTH languages are standard with TFORTH. These features include:

- ■ *Advanced Math Package*
- ■ *Line Editor*
- ■ *Macro Assembler*
- ■ *Re-Entrant Code*
- ■ *Super Graphics Capabilities*
- ■ *Sophisticated User Functions*
- ■ *140 Page User's Manual*
- ■ *Virtual memory*
- ■ *Interpreter*
- ■ *Compiler*
- ■ *Produces CMD Files*
- ■ *Expandable*
- ■ *And many, many other features*

TFORTH from SIRIUS comes on diskette complete for the TRS-80® with as little as 16K of memory and a single Disk Drive.
TFORTH **$129.95**

PRIAM Hard Disks Now Available from SIRIUS SYSTEMS!



PRIAM's high-performance, low-cost Winchester disc drives speed up throughput and expand data storage from 20 megabytes to 154 megabytes. And a single controller can be used to operate 14-inch-disc drives with capacities of 33, 66, or 154 megabytes or floppy-disc-size drives holding 20 and 34 megabytes. So it's easy to move up in capacity, or reduce package size, without changing important system elements or performance.

- ■ *Fast, Linear Voice Coil Positioning*
- ■ *10 ms track-to-track positioning*
- ■ *Fully servoed head positioning*
- ■ *Dedicated servo tracks*
- ■ *DC Power required only!*
- ■ *Simple, parallel Interface*
- ■ *Optional SMD Interface*
- ■ *50 ms Average Positioning time*
- ■ *90 ms Maximum Positioning Time*
- ■ *6.4 ms Average Latency*

THE PRIAM LINEUP

Model/Disc Size	Capacity	Size	Weight	Price
DISKOS 2250(14")	22Mbytes	7" × 17" × 20"	33lbs.	**$2495**
DISKOS 3350 (14")	33Mbytes	7" × 17" × 20"	33 lbs.	**$2995**
DISKOS 6650 (14")	66 Mbytes	7" × 17" × 20"	33 lbs.	**$3749**
DISKOS 15450 (14")	154 Mbytes	7" × 17" × 20"	33 lbs.	**$4695**
DISKOS 2050 (8")	20 Mbytes	4 62" × 8.55" × 14.25"	20 lbs.	**$2995**
DISKOS 3450 (8")	34 Mbytes	4.62" × 8.55" × 14.25"	20 lbs.	**$3745**
DISKOS 1070	10.6 Mbytes	floppy-size	(low)	**$2195**

Optional SMD interface available for $150.

SIRIUS SYSTEMS Hard Disk Enclosures include all internal cabling, a power supply, case and a fan for reliable operation.

14" DISKOS DRIVE CASE..............$375.00
8" DISKOS DRIVE CASE..............265.00

STATE-OF-THE-ART DISK DRIVES

QUME® DataTrak 8
8" Disk Drive
DOUBLE SIDED! DOUBLE DENSITY!

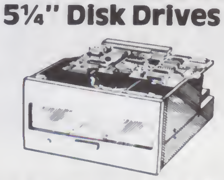
High performance Double Sided Disk 8" Disk Drive ■ Single or Double Density ■ Door Lock and Write Protect INCLUDED! ■ Negative DC Voltage not required ■ Low Power Operation

- ■ *FAST! 3ms track-to-track access*
- ■ *Low friction and minimum wear*
- ■ *Superior Head Load Dynamics*

QUME DataTrak 8 ....**$574.95**

QUME Technical Manual **$12.95**
Connector Set #3 (AC, DC, & Card Edge) **$10.95**
Connector Set #4 (AC and DC) **$2.95**

MPI 51/52 & 91/92 5¼" Disk Drives



- ■ *Fast! 5ms track-to-track access*
- ■ *Exclusive Pulley-Band Design*
- ■ *Unique Door/Ejector Mechanism*
- ■ *Reliable 1 1/2% Speed Stability*
- ■ *Single/Double Density Operation*
- ■ *Industry/ANSI Standard Interface*

MPI 51 (Single Head/40 tracks)
125K/250K Bytes Single/Double Density** **$259.95**
MPI52 (Dual Head/80 tracks (40/side))
250K/500K Bytes Single/Double Density** **$349.95**
MPI 91 (Single Head/80 tracks)
250K/500K Bytes Single/Double Density** **$399.95**
MPI 92 (Dual Head/160 tracks (80/side))
500K/1000K Bytes Single/Double Density** **$524.95**
MPI Technical Manual **$12.95**
**Unformatted data storage

TO ORDER CALL (615) 693-6583

Phone Orders Accepted 9AM-7PM (EST) Mon-Fri

We accept **MC, VISA, AE, COD** (requires Certified Check, Cashier's Check or Cash) and **Checks** (personal checks require 14 days to clear). **SHIPPING AND HANDLING:** $7.00 per Floppy Disk Drive or 80+ Module ■ 5% for other items (any excess will be refunded) ■ **Foreign Orders** add 10% for Shipping & Handling. Payment in U.S. currency ■ Tennessee residents add 6% Sales Tax ■ **VOLUME DISCOUNTS AVAILABLE**

✓67

SIRIUS SYSTEMS
7528 Oak Ridge Highway
Knoxville, Tennessee 37921

✓Reader Service—see page 178

CLASSIFIEDS

Classified advertisements are intended for use by persons desiring to buy, sell or trade used computer equipment. No commercial ads are accepted.

Two sizes of ads are available. The $5 box allows up to 5 lines of about 35 characters per line, including spaces and punctuation. The $10 box allows up to 10 lines. Minimize use of capital letters to save space. No special layouts allowed. Payment is required in advance with ad copy. We cannot bill or accept credit.

Advertising text and payment must reach us 60 days in advance of publication (i.e., copy for March issue, mailed in February, must be here by Jan. 1). The publisher reserves the right to refuse questionable or inapplicable advertisements. Mail copy with payment to: **Classifieds, Kilobaud Microcomputing, Peterborough NH 03458.** Do not include any other material with your ad as it may be delayed.

Wanted: used cassette software for Apple II & books on computing. Send list for my offer. Ray Gabriel, PO Box 175, Madison, OH 44057.

For Sale: BASF C-10 computer cassettes. Low noise, high output, shipped prepaid UPS .75 each. A. Ringel W6KVO, 1118 Raintree Circle, Culver City, CA 90230.

For Sale: Mits Altair 88-DCDD floppy-disk drive and controller. Recently aligned and all ECOs installed. With all manuals and software including Altair Disk BASIC, $700. Diablo printer with stand, print wheels, ribbons, paper and full documentation, $1250. Will deliver within 75 miles. Call to arrange demo. Alan Frisbie, 3786 E. Mountain View, Pasadena, CA 91107, 213-351-2351 days, 796-7872 evenings.

For Sale: 12-inch B/W monitor. Composite video input, excellent resolution. Never used, in original carton. $125. Mark Simpson, 900 Lake St., Apt. 8, Houghton, MI 49931. 906-482-8856.

Commodore PET. 16K RAM expandable to 32K on-board. 14K ROM w/machine-language monitor & improved BASIC. 8K ROM expansion sockets on-board. Typewriter keyboard and number pad. C2N external cassette recorder. Technical manuals plus "PET & IEEE Bus" and "PET Personal Computer Guide" books. $750, plus shipping. Greg after 5 PM at 513-851-1936.

North Star 2 system. Double density, 48K, with D.C. Hayes modem. 2 serial & 1 parallel ports, Sorac IQ 140 terminal, Centronics 779 tractor printer, includes apx $2000 in business software, 6 months old, in org boxes, $4895 or best offer. Also one D.C. Hayes modem 100, $325. Call Don, 615-526-7651.

Software (personal) for sale. Sold system must sell at super reduced prices. Dozens of CP/M, languages, business s/w (G/L, A/R, A/P, etc.). Send SASE for complete list and prices. Dee K. Moser, Box 638, Great Bend, KS 67530.

For Sale: Heath H9 video term., $200; Cherry Pro Keybd w/encl., internal pwr sup., $130. H9 prints lwr case input from Cherry/computer. H9 avail w/o mod. I ship. George Sullivan c/o Dr. P. Petcher, Chatom, AL 36518. 205-847-2262.

Diablo Hytype I Mod 1200 daisy wheel printer; new pin feed fric. platen & print wheel; white 'RO' cover; new Boschert switch supply; interface cable, service manual—$1000. Barry: 415-228-6211.

For Sale or trade for Heath H-89 or H8 sys. Kenwood ham equ. TS520S + freq readout + AT200 tuner + 6 and 2m xverters and VFO. All service manuals and cables. Edward H. Hill, Sr., 317-545-8886.

F&D 64 × 16 VDB-1 video board for SWTP 6800 bus, assembled, socketed, not tested with complete documentation—$75. Heath H-8, 8K static memory board assembled/tested without memory—$50. Also Mostek VAB-2 (F8) video board 64 × 16 generates 128 ASCII chr's, parallel keyboard input, TTL/RS-232 serial out/in, composite video requires +5V @ .75A, with complete documentation—$150. Paul Ramos, 100 Middle St., Woburn, MA 01801. 617-935-3758 after 6 PM.

Heath H-8: H9; excellent condition: (3) 8K memory boards, H-8-5 I/O board. All manuals. Estate sale price very cheap. All Benton Harbor software plus BASIC Programming Course. Leon Fernald, Box 1027, Kingshill, St. Croix, V.I. 00850. 809-778-3845.

For Sale: Netronics Elf II in steel enclosure, 2-4K static RAM boards, giant board, color graphics/music board, expansion P.S., rf modulator, fully socketed, all Netronics data, extras. All for $425. Harold E. Kautz, Jr., 1115 E. Caracas Ave., Hershey, PA 17033. 717-534-2642, after 6 PM.

For Sale: Z-80 computer: 4K RAM, monitor, kybd, readout, cassette intf., EPROM programmer, 16 I/O ports, S-100, $325. State logic analyser: 16-bit input, 5 MHz, 16 LED, 4 hex readout, edge clocking, forward and backward stepping, $390. 6809 computer: monitor, editor/assembler, RS-232, 16 I/O ports, timer, cassette intf, $400. Gopaul, 3 Weymouth Ave., Toronto, Ont., M4C-1R6, Canada. 416-698-4405.

Super electronics shop for $800! Instruments, texts (all Kilobaud natch), components, etc., everything. Owner can't use the approx. $1800 tax credit. A roomful of prime at choice price! SASE for inv. $1 for pix. 334 Riverside Drive, Palm Beach Gardens, FL 33410.

Find it Here!

MICRO QUIZ

(from page 20)

Answer: 9

At statement 70:

X(1) = 4   X(4) = 5
X(2) = 9   X(5) = 1
X(3) = 2   X(6) = 8

Statement 90 changes the value of V only when it reaches an element of X greater than V's current contents. Thus, V contains the largest element of X.

✓ Reader Service—see page 178

DEALER DIRECTORY

Phoenix, AZ
Software Specialists! Authorized dealers in quality software packages. Experts provide comprehensive product demonstrations, outstanding support and free consultation. Resident CP/M swami. Arizona's finest service department. **The Denver-Ridge Group, Suite 104, 8900 North Central Avenue, Phoenix, AZ 85020, 944-8371.**

El Monte, CA
Ohio Scientific specialist in the San Gabriel Valley serving greater Los Angeles. Full product line on display. Specializing in business computers. In-house service. Custom programming. Terminals. Printers. Open Mon–Sat, 9 AM–7 PM. **Computer & Video, 3380 Flair Dr., Suite 207, El Monte, CA 91731, 572-7292.**

San Jose, CA
Bay area's newest computer software store. Featuring Instant Software for the TRS-80, magazines, books. **Shaver Radio, 1378 S. Bascom Ave., San Jose, CA 95128, 998-1103.**

San Jose, CA
New and used computer products—specializing in S-100 boards, printers, drives, chassis and complete systems, as well as supplies and parts —Imsai, Tandon, Diablo—5000 sq. ft. **W/W Component Supply, Inc., 1771 Junction Ave., San Jose, CA 95112.**

Denver, CO
Tantalizing software for your microcomputer. We have software applications for both home and business. Ask for prices on diskettes, tapes, custom forms, checks, etc. **The Software Gourmet, 1111 South Pearl St., Denver, CO 80210, 777-8890.**

Sarasota, FL
Dynabyte computer systems, Hazeltine and NEC, Word-Star, Structured Systems accounting. Consulting, training, sales, service. **Glisco, Inc., 4001 Roberts Point Rd., Sarasota, FL 33581, 349-0200.**

Sarasota, FL
Real-Time/Industrial Software Engineering Consultants. Top-down design and structured techniques utilized. **Advanced Computer Concepts, Inc., PO Box 15710, Sarasota, FL 33579.**

Tampa, FL
Apple Computer sales and service. S-100 boards from SSM, Godbout, Thinker Toys, California Computer Systems. Computer books and magazines. **AMF Microcomputer Center, Inc., 11158 N. 30th Street, Tampa, FL 33612, 971-4072, 977-0708.**

Aurora, IL
Microcomputer systems for home or business; peripherals, software, books & magazines. Apple, Hewlett-Packard, North Star, Cromemco systems. IDS-440G printer w/Apple graphics, new HP-85 & HP calculators. **Farnsworth Computer Center, 1891 N. Farnsworth Ave., Aurora, IL 60505, 851-3888.**

Chicago, IL
Computer Hardware/Specialists for home and business. Largest selection of computer books, magazines and copyrighted software in Illinois. Experienced factory-trained service department. Feature Apple, Alpha Microsystems and Hewlett-Packard calculators and accessories. **Data Domain of Schaumburg, 1612 E. Algonquin Rd., Schaumburg, IL 60195, 397-8700.**

Chicago, IL
Brand new lowest prices, never undersold, postpaid in USA—Teletype 43 keyboard printers, Okidata & Integral Data printers, SS-50 bus computers, peripherals & business software. **Data Mart, 914 East Waverly Street, Arlington Heights, IL 60004, 398-8525.**

Dial a Dealer

Herington, KS
Hardware support. Maintenance and service for all microcomputers and peripherals. Kits assembled or debugged. Radio Shack (mods OK) repaired. Quality work, fast turnaround and reasonable cost. **Prairie Micro Clinic, Box 325, Herington, KS 67449, 258-2179.**

Garden City, MI
Books, magazines, hardware and software for Apple, North Star, TRS-80 and PET. **Computer Center, 28251 Ford Rd., Garden City, MI 48135, 425-2470.**

Hannibal, MO
Ohio Scientific products, modifications, service, software. 8" disk for C1p, C4p. Process control specialist. **E&I Technical Service, 5300 Paris Gravel Road, Hannibal, MO 63401, 248-0084.**

Portland, OR
Ohio Scientific specialists for business and personal computers. Local service. Terminals, printers, custom programming. Full OSI product line on display! 10 AM to 6 PM M–F. **Fial Computer, 11266 SE 21st Ave., Milwaukie, OR 97222, 654-9574.**

Dealers: Listings are **$15** per month in prepaid quarterly payments, or one yearly payment of **$150**, also prepaid. Ads include 25 words describing your products and services plus your company name, address and phone. (No area codes or merchandise prices, please.) Call Marcia at 603-924-7138 or write *Kilobaud Microcomputing*, Ad Department, Peterborough, NH 03458.

Who Says Programmers Are Dull?

Here's a chance to display your creative, imaginative and artistic talents!
Enter the first **Microcomputing Graphics Contest.** You can win $100 for the best microcomputer-generated graphics display in any of the following categories:

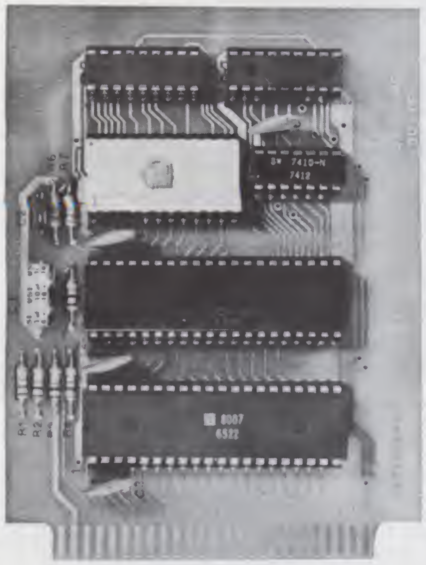
- **Color video**
- **Black and white video**
- **Printer hard copy**
- **Plotter hard copy**

Second prize is a $50 certificate for books from our Book Nook section. Third prize is a one-year subscription to *Microcomputing*.

Limit: two submissions per category.
Deadline: April 1, 1981.
Winners will be selected by a panel of judges from the computer science and art departments of Keene State College (Keene, N.H.) and Franklin Pierce College (Rindge, N.H.).

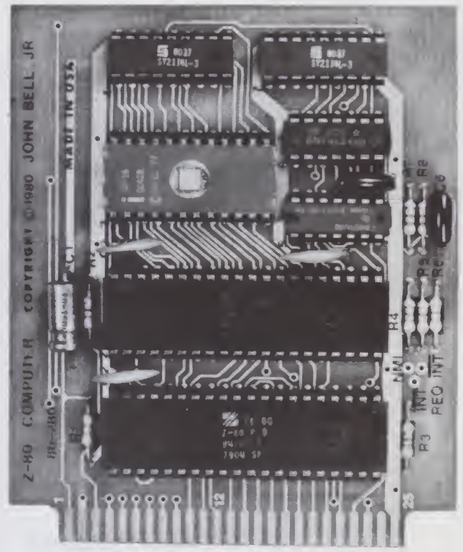
- The following information must accompany each entry: Your name, address and phone number; the category of the entry; title of the entry; description of the hardware; name of any commercial software used in the creation of your work.
- For video categories, send 5×7 inch glossy prints of the artwork as it appears on your video display.
- For the hard-copy categories, send full-size originals.
- While you don't need to submit programs with your entries, they must be available on request.
- No minis or mainframes allowed.
- Entries become the property of *Microcomputing* magazine, and cannot be returned.
- Address entries to **Graphics Contest, Kilobaud Microcomputing Editorial Dept., Peterborough, NH 03458.**

6502, Z80, 8080 AND 8085 USERS



JOHN BELL ENGINEERING'S 6502 AND Z80 MICROCOMPUTERS ARE DEDICATED COMPUTERS DESIGNED FOR **CONTROL FUNCTIONS.**

THESE BOARDS FEATURE:
- 2048 BYTES EPROM
- 1024 BYTES RAM
- ALL BOARDS INCLUDE COMPLETE DOCUMENTATION
- 50 PIN CONNECTOR INCLUDED IN KITS AND ASSEMBLED UNITS
- 2716 AVAILABLE SEPARATELY



JOHN BELL ENGINEERING'S 6502 MICROCOMPUTER, 2716 EPROM PROGRAMMER AND APPLE II PARALLEL INTERFACE PLUS THE APPLE II MICROCOMPUTER — **A COMPLETE DEVELOPMENT SYSTEM.**

6502 MICROCOMPUTER FEATURES:
- 1024 BYTES RAM
- 2048 BYTES EPROM
- USES ONE 6522 VIA (DOC. INCL.)
  - 2 8 BIT BIDIRECTIONAL I/O PORTS
  - 2 16 BIT PROGRAMMABLE TIMER/COUNTERS
  - SERIAL DATA PORT
  - LATCHED I/O WITH HANDSHAKING LOGIC
  - TTL AND CMOS COMPATIBLE

80-153 **ASSEMBLED $110.95**
BARE BOARD $24.95 KIT $ 89.95

JOHN BELL ENGINEERING'S NEW Z80 MICROCOMPUTER FEATURES:
- Z80 CPU SOFTWARE COMPATIBLE WITH Z80, 8080 AND 8085 MICROPROCESSORS
- 2048 BYTES EPROM
- 1024 BYTES RAM
- SINGLE 5V POWER SUPPLY AT 300MA
- CLOCK FREQUENCY IS 2MHZ, RC CONTROLLED
- Z80 PIO (DOC. INCL.)
- 2 8 BIT BIDIRECTIONAL I/O PORTS
- LATCHED I/O WITH HANDSHAKING LOGIC
- TTL AND CMOS COMPATIBLE

80-280 **ASSEMBLED $129.95**
BARE BOARD $29.95 KIT $119.95

USE YOUR 6502 OR Z80 MICROCOMPUTER TO CONTROL EVERYTHING!
- YOUR HOME SECURITY SYSTEM
- HEAT CONTROL
- LIGHT CONTROL
- SOLAR HEATING AND POWER SYSTEMS
- AUTOMATIC CONTROL OF TAPE RECORDERS
- TRAFFIC LIGHT CONTROL
- IRRIGATION SYSTEMS
- AUTOMATIC CONTROL OF VIDEO RECORDERS
- ROBOT CONTROL
- AUTOMATIC DIALER
- AUTOMATED SLIDE SHOW CONTROL
- COMMUNICATION SYSTEMS FOR THE DISABLED
- THE WORLD

JOHN BELL ENGINEERING ✓99

ALL PRODUCTS ARE AVAILABLE FROM: **JOHN BELL ENGINEERING**
P.O. BOX 338 • REDWOOD CITY, CA 94064 • (415) 367-1137

VISA

*SEND FOR OUR COMPLETE CATALOG! ADD 6% SALES TAX IN CALIFORNIA. ADD 5% FOR SHIPPING AND HANDLING.

MASTER CHARGE

✓ *Reader Service—see page 178*

Skyles Electric Works

66

BASIC Programmer's, Toolkit™, Disk-O-Pro™, Command-O™

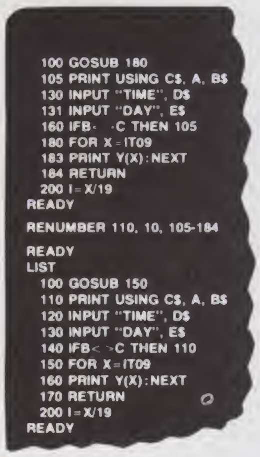
For CBM™ Owners Who Want More Fun And Fewer Errors with Their Programs

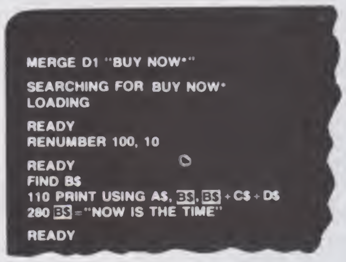
Here are nineteen commands you'll need, on a single chip you can install in two minutes without tools, **on any CBM or CMB system.** 4KB of ROM firmware on each chip with a collection of machine language programs available from the time you turn on your PET to the time you shut it off.

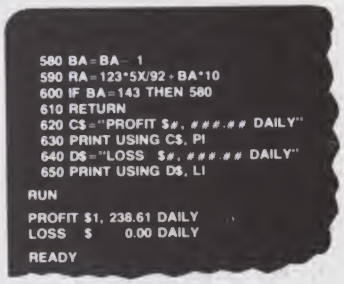
For CBM 8016 and 8032; BASIC 4.0

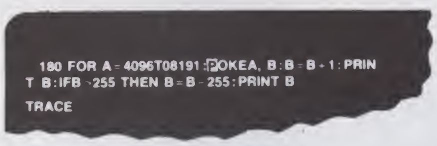
BASIC Programmers Command-O™

AUTO[ed] **DUMP**[ed] **DELETE**[ed] **FIND**[ed] (improved) **HELP**[ed] **KILL**[ed] **OFF**[ed] **TRACE**[ed] (improved) **RENUMBER**[ed] (improved) **INITIALIZE**[BS] **MERGE**[BS] **MOVE**[BS] **EXECUTE**[BS] **SCROLL**[ed] **OUT**[ed] **SET**[ed] **SEND**[BS] **PRINT USING**[BS] **BEEP**[BS]









PRICES:

BASIC Programmers **Toolkit™** (chip only)	$40.00
BASIC Programmers **Disk-O-Pro™** (chip only)	$75.00
BASIC Programmers **Command-O™** (chip only)	$75.00
Interface boards (needed sometimes)	$20.00-$50.00
Instruction Manual (with redeemable $5.00 coupon)	$5.00

Shipping and handling $2.50 USA/Canada, $10.00 Europe/Asia

California residents please add 6% or 6-1/2% sales tax as required

*Reserve your **Disk-O-Pro, Command-O** today*

Toolkit™ immediate delivery, Disk-O-Pro delivery in December, Command-O delivery in January

VISA, MASTERCHARGE ORDERS CALL (800) 538-3083 (except California residents)

CALIFORNIA ORDERS PLEASE CALL (408) 257-9140

Skyles Electric Works

231 E South Whisman Road
Mountain View, CA 94041
(415) 965-1735

ORDER ENTRY/ACCOUNTS RECEIVABLE SYSTEMS

Fyrnetics President, Larry Larsen, and Controller, Dennis Turek, considering some of the features of their new security system.

THE COMPANY:
FYRNETICS, INC.
1021 DAVIS ROAD
ELGIN, ILLINOIS 60120
312-742-0282

Fyrnetics, Inc. of Elgin, Illinois has grown from a private manufacturer of ionization smoke detectors for companies, such as Sears & Roebuck, to a full line manufacturer of wireless security products which are sold through a worldwide network of dealers and distributors. Since saturation of the residential security market is estimated to be less than 3% in the United States, their major marketing effort has been directed toward distribution through dealers who demonstrate and sell electronic products to the consumer. This marketing approach generated such a high volume of sales orders and invoices that it became necessary to install a computer system in 1977 in order to handle the multi-million dollar annual sales volume.

"Fyrnetics initially installed a Wang 2200 computer system, equipped with seven CRT terminals, four 10 megabyte hard disk drives, and two line printers at a monthly lease cost of over $4000.00. By late 1979, the company had invested nearly six man-years in our own software development", comments Fyrnetics president, Larry Larsen "As a part of a cost cutting program, we decided to replace the Wang equipment with a microcomputer system. We reviewed all available micro systems and concluded that the MSI hardware and software packages were best suited to our needs."

The modern production facility at Fyrnetics, Inc. where wireless home security products are designed and manufactured. The company also has production facilities in Hong Kong where larger quantities of their products are manufactured.

"We were particularly pleased with the system-generation capability of the MSI business packages, which allowed us to utilize our existing continuous form sales orders, invoices, and packing lists. The requirements for our computer system were rather demanding since we had large customers such as Montgomery Wards and Wicks, each having up to 300 stores. Each store is treated as an individual customer during the order entry and shipping process. However, payment is made from a central accounting office with many stores on a single check. Our system had to allow us to properly credit the payment to many different store locations and invoices. This feature was a part of the MSI Accounts Receivable software package. **Our accounts receivable system handles over 750 regular customers with over 3000 open invoices and 10,000 transactions per month for us."**

The LIFESAVER line of wireless home security products manufactured by Fyrnetics, Inc. Anyone desiring more information on this interesting product line should contact Fyrnetics, Inc. at the above address.

CONSIDERATIONS FOR THE SELECTION OF ACCOUNTS RECEIVABLE SYSTEMS

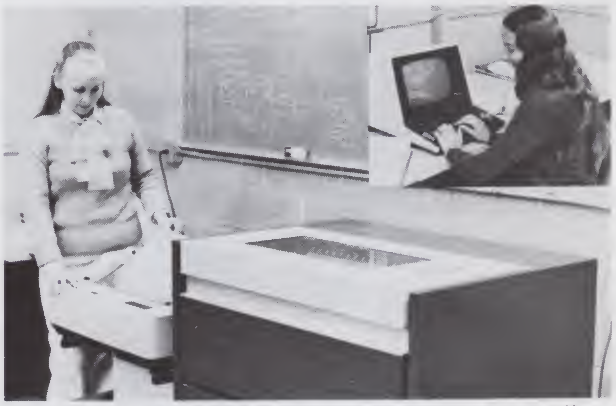
The selection of business computer systems today involves the careful consideration of many different factors. Even though the cost of computer systems has dropped substantially, we considered the selection process to be highly critical to us because of the tremendous need for a highly reliable computer system in our daily operations. Due to our high volume of sales transactions we were highly dependent upon the system for order processing and for information. We considered the following issues to be key to our selection of the MSI system:

LARGE DATA BASE - The processing of over 700 orders per month, with 3000 open invoices and 5000 active statement items required that we have easy and efficient on-line access to our large data base. The MSI system provides a large selection of data reports for open orders, backorders, invoices, credit memos, as well as customer statements and account status information.

EFFICIENT PAPERWORK FLOW - The processing of our large volume of sales orders required an efficient system for printing sales orders, packing lists, invoices, and customer statements. The MSI system offers a convenient system generation program which allows the use of any desired format for pre-printed continuous forms. In addition, packing lists and customer invoices are generated automatically as sales orders are processed.

GENERAL LEDGER TIE-IN - Due to the large volume of individual invoices and cash receipts, we required an automatic posting procedure for our general ledger programs in order to minimize the data entry process. The MSI system offers a complete general ledger program package which links automatically to the other business program modules. All invoices, as well as cash receipts, are automatically written to the general ledger posting files from which individual journals are created. This procedure insures the generation of balanced journals and greatly reduces the time requirement for generation of monthly income statement and balance sheets.

SYSTEM INTEGRATION - The MSI system is fully integrated. The order entry system is linked to inventory for correct pricing, description of items on order. The inventory system is also linked to general ledger to allow different categories of products sold to be automatically posted to the correct sales accounts. The MSI inventory system provides complete cost accounting information for both labor and material. **The MSI programs provide the big machine capability that we need and yet provide the flexiblilty that we desire.**



The MSI computer system drives two line printers at Fyrnetics, Inc.

The MSI computer system at Fyrnetics, Inc. employs 10 megabyte hard disk drives to contain the large on-line data base.

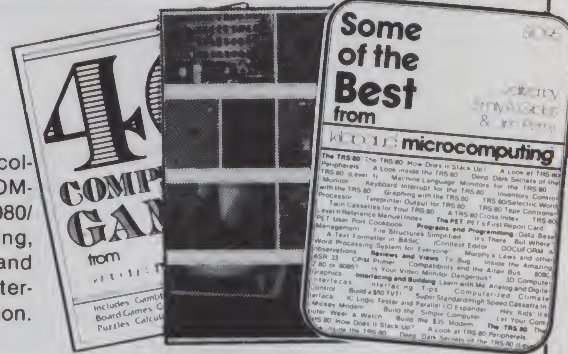
SUPPORT - The availability of source listings for all of the MSI business software was an added incentive to select the MSI system. This has allowed us to make some specialized enhancements to our programs easily. **MSI really delivered for us allowing the replacement of an expensive WANG 2200 system with a comparable MSI system at a fraction of the cost.**

If you would like to have more information on MSI business computer systems, call or write, **Midwest Scientific Instruments, Inc., 220 W. Cedar, Olathe, KS 66061, 800-255-6638, Telex 42525(MSI A OLAT).**

MSI ..... Helping to make your business run better.

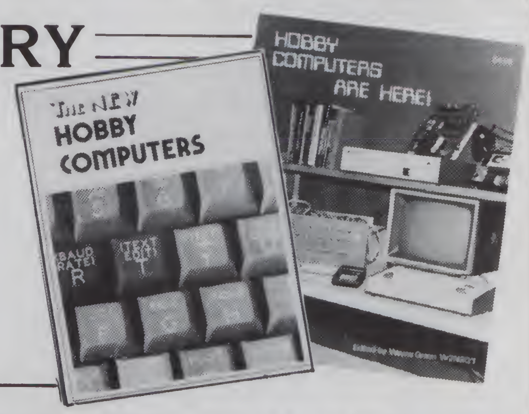
kb microcomputing book nook

3 NEW BOOKS



- **SOME OF THE BEST FROM KILOBAUD/MICROCOMPUTING**—BK7311—A collection of the best articles that have recently appeared in Kilobaud/MICROCOMPUTING. Included is material on the TRS-80 and PET systems, CP/M, the 8080/8085/Z80 chips, the ASR-33 terminal. Data base management, word processing, text editors and file structures are covered too. Programming techniques and hardcore hardware construction projects for modems, high speed cassette interfaces and TVTs are also included in this large format, 200 plus page edition. $10.95.*
- **UNDERSTANDING AND PROGRAMMING MICROCOMPUTERS**—BK7382—A valuable addition to your computing library. This two part text includes the best articles that have appeared in 73 and Kilobaud Microcomputing magazines on the hardware and software aspects of the new microcomputing hobby. Well known authors and well structured text helps the reader get involved in America's fastest growing hobby. $10.95*
- **40 COMPUTER GAMES**—BK7381—Forty games in all in nine different categories. Games for large and small systems, and even a section on calculator games. Many versions of BASIC used and a wide variety of systems represented. A must for the serious computer gamesman. $7.95*

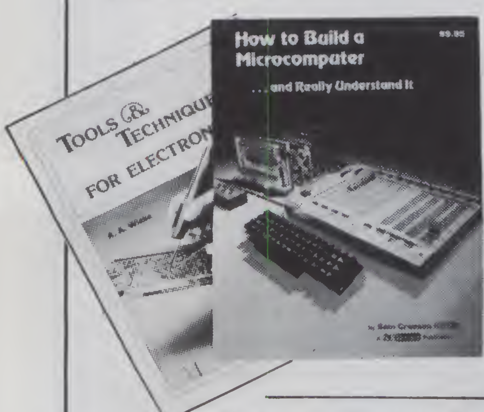
INTRODUCTORY



- **HOBBY COMPUTERS ARE HERE!**—BK7322—If you (or a friend) want to come up to speed on how computers work . . . hardware and software . . . this is an excellent book. It starts with the fundamentals and explains the circuits, and the basics of programming. This book has the highest recommendations as a teaching aid for newcomers. $4.95.*
- **THE NEW HOBBY COMPUTERS**—BK7340—This book takes it from where "HOBBY COMPUTERS ARE HERE!" leaves off, with chapters on Large Scale Integration, how to choose a microprocessor chip, an introduction to programming, low cost I/O for a computer, computer arithmetic, checking memory boards . . . and much, much more! Don't miss this tremendous value! Only $4.95.*

INTRODUCTION TO MICROCOMPUTERS (VOL. 0→III)

- **AN INTRODUCTION TO MICROCOMPUTERS, VOL. 0**—BK1130—The Beginner's Book—Written for readers who know nothing about computers—for those who have an interest in how to use computers—and for everyone else who must live with computers and should know a little about them. The first in a series of 4 volumes, this book will explain how computers work and what they can do. Computers have become an integral part of life and society. During any given day you are affected by computers, so start learning more about them with Volume 0. $7.95.*
- **VOL. I**—BK1030—**2nd Edition completely revised.** Dedicated to the basic concepts of microcomputers and hardware theory. The purpose of Volume I is to give you a thorough understanding of what microcomputers are. From basic concepts (which are covered in detail), Volume I builds the necessary components of a microcomputer system. This book highlights the difference between minicomputers and microcomputers. $12.99.*
- **VOL. II**—BK1040 (with binder)—Contains descriptions of individual microprocessors and support devices used only with the parent microprocessor. Volume II describes all available chips. $31.99*
- **VOL. III**—BK1133 (with binder)—Contains descriptions of all support devices that can be used with any microprocessor. $21.99*



- **HOW TO BUILD A MICROCOMPUTER—AND REALLY UNDERSTAND IT**—BK7325—by Sam Creason. The electronics hobbyist who wants to build his own microcomputer system now has a practical "How-To" guidebook. This book is a combination technical manual and programming guide that takes the hobbyist step-by-step through the design, construction, testing and debugging of a complete microcomputer system. Must reading for anyone desiring a true understanding of small computer systems. $9.95.*
- **TOOLS & TECHNIQUES FOR ELECTRONICS**—BK7348—by A. A. Wicks is an easy-to-understand book written for the beginning kit builder as well as the experienced hobbyist. It has numerous pictures and descriptions of the safe and correct ways to use basic and specialized tools for electronic projects as well as specialized metal working tools and the chemical aids which are used in repair shops. $4.95.*

*Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to Kilobaud *Microcomputing* Book Department • Peterborough NH 03458. Be sure to include check or detailed credit card information. No C.O.D. orders accepted. All orders add $1.00 handling. Please allow 4-6 weeks for delivery. Questions regarding your order? Please write to Customer Service at the above address.

"PRICES SUBJECT TO CHANGE WITHOUT NOTICE"

FOR TOLL FREE ORDERING CALL 1-800-258-5473

kb microcomputing book nook

SPECIAL INTERESTS

new!

new!

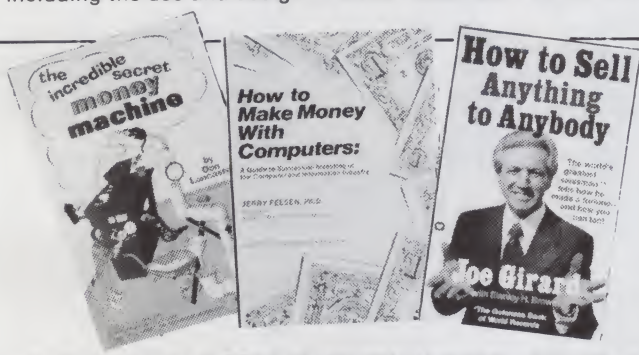
● **MICROSOFT BASIC DECODED AND OTHER MYSTERIES**—BK1186—by James Farvour. From the company that brought you *TRS-80 DISK AND OTHER MYSTERIES*! Contains more than 6500 lines of comments for the disassembled Level II ROMs, six additional chapters describing every BASIC subroutine, with assembly language routines showing how to use them. Flow charts for all major routines give the reader a real insight into how the interpreter works. $29.50 (Available after December 20th).

● **THE CP/M HANDBOOK (with MP/M)**—BK1187—by Rodnay Zaks. A complete guide and reference handbook for CP/M—the industry standard in operating systems. Step-by-step instruction for everything from turning on the system and inserting the diskette to correct user discipline and remedial action for problem situations. This also includes a complete discussion of all versions of CP/M up to and including 2.2, MP/M and CDOS. $13.95.

● **TRS-80 DISK AND OTHER MYSTERIES**—BK1181—by Harvard C. Pennington. This is the definitive work on the TRS-80 disk system. It is full of detailed "How to" information with examples, samples and in-depth explanations suitable for beginners and professionals alike. The recovery of one lost file is worth the price alone. $22.50.*

● **INTRODUCTION TO TRS-80 GRAPHICS**—BK1180—by Don Inman. Dissatisfied with your Level I or Level II manual's coverage of graphics capabilities? This well-structured book (suitable for classroom use) is ideal for those who want to use all the graphics capabilities built into the TRS-80. A tutorial method is used with many demonstrations. It is based on the Level I, but all material is suitable for Level II use. $8.95.*

● **MICROPROCESSOR INTERFACING TECHNIQUES**—BK1037—by Austin Lesea & Rodnay Zaks—will teach you how to interconnect a complete system and interface it to all the usual peripherals. It covers hardware and software skills and techniques, including the use and design of model buses such as the IEEE 488 or S100. $15.95.*



MONEYMAKING

● **HOW TO MAKE MONEY WITH COMPUTERS**—BK1003—In 10 information-packed chapters, Jerry Felsen describes more than 30 computer-related, money-making, high profit, low capital investment opportunities. $15.00.*

● **HOW TO SELL ANYTHING TO ANYBODY**—BK7306—According to *The Guinness Book of World Records*, the author, Joe Girard, is "the world's greatest salesman." This book reveals how he made a fortune—and how you can, too. $2.25.*

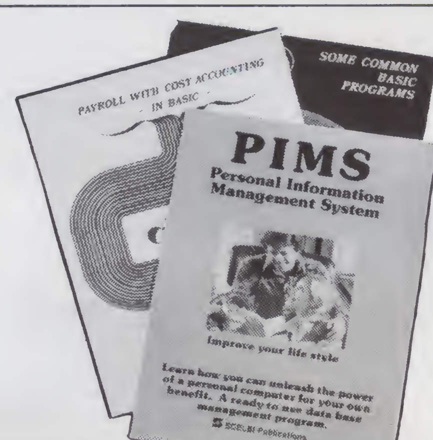
● **THE INCREDIBLE SECRET MONEY MACHINE**—BK1178—by Don Lancaster. A different kind of "cookbook" from Don Lancaster. Want to slash taxes? Get free vacations? Win at investments? Make money from something that you *like* to do? You'll find this book essential to give you the key insider details of what is really involved in starting up your own money machine. $5.95.*

BUSINESS

● **PAYROLL WITH COST ACCOUNTING—IN BASIC**—BK1001—by L. Poole & M. Borchers, includes program listings with remarks, descriptions, discussions of the principle behind each program, file layouts, and a complete user's manual with step-by-step instructions, flowcharts, and simple reports and CRT displays. Payroll and cost accounting features include separate payrolls for up to 10 companies, time-tested interactive data entry, easy correction of data entry errors, job costing (labor of distribution), check printing with full deduction and pay detail, and 16 different printed reports, including W-2 and 941 (in **CBASIC**). $20.00.*

● **SOME COMMON BASIC PROGRAMS**—BK1053—published by Adam Osborne & Associates, Inc. Perfect for non-technical computerists requiring ready-to-use programs. Business programs, plus miscellaneous programs. Invaluable for the user who is not an experienced programmer. All will operate in the stand-alone mode. $14.99 paperback.

● **PIMS: PERSONAL INFORMATION MANAGEMENT SYSTEM**—BK1009—Learn how to unleash the power of a personal computer for your own benefit in this ready-to-use data-base management program. $11.95.*



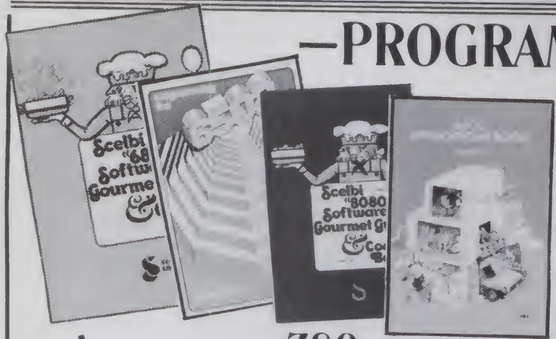
"PRICES SUBJECT TO CHANGE WITHOUT NOTICE"

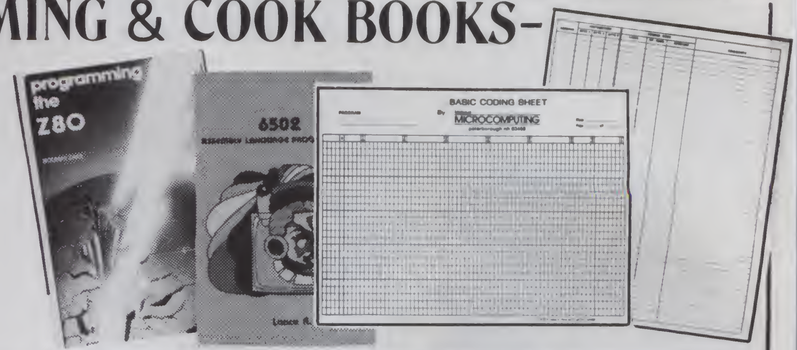
*Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to Kilobaud *Microcomputing* Book Department • Peterborough NH 03458. Be sure to include check or detailed credit card information. No C.O.D. orders accepted. All orders add $1.00 handling. Please allow 4-6 weeks for delivery. Questions regarding your order? Please write to Customer Service at the above address.

FOR TOLL FREE ORDERING CALL 1-800-258-5473

kb microcomputing book nook

—PROGRAMMING & COOK BOOKS—





—Z80—

new!

● **INSIDE LEVEL II** — BK1183 — For machine language programmers! This is a comprehensive reference guide to the Level II ROMs, allowing easy utilization of the sophisticated routines they contain. It concisely explains set-ups, calling sequences, variable passage and I/O routines. Part II presents an entirely new composite program structure which unloads under the SYSTEM command and executes in both BASIC and machine code with the speed and efficiency of a compiler. Special consideration is given to disk systems. $15.95.*

● **PROGRAMMING THE Z-80** — BK1122 — by Rodnay Zaks. Here is assembly language programming for the Z-80 presented as a progressive, step-by-step course. This book is both an educational text *and* a self-contained reference book, useful to both the beginning and the experienced programmer who wish to learn about the Z-80. Exercises to test the reader are included. $14.95.*

● **Z-80 ASSEMBLY LANGUAGE PROGRAMMING** — BK1177 — by Lance A Leventhal. This book thoroughly covers the Z80 instruction set, abounding in simple programming examples which illustrate software development concepts and actual assembly language usage. Features include Z80 I/O devices and interfacing methods, assembler conventions, and comparisons with 8080A/8085 instruction sets and interrupt structure. $16.99.*

● **Z-80 SOFTWARE GOURMET GUIDE AND COOKBOOK** — BK1045 — by Nat Wadsworth. Scelbi's newest cookbook! This book contains a complete description of the powerful Z-80 instruction set and a wide variety of programming information. Use the author's ingredients including routines, subroutines and short programs, choose a time-tested recipe and start cooking! $15.95.*

—6502—

● **PROGRAMMING THE 6502 (Third Edition)**—BK1005— Rodnay Zaks has designed a self-contained text to learn programming, using the 6502. It can be used by a person who has never programmed before, and should be of value to anyone using the 6502. The many exercises will allow you to test yourself and practice the concepts presented. $12.95.*

● **6502 APPLICATIONS BOOK** — BK1006 — Rodnay Zaks presents practical-application techniques for the 6502 microprocessor, assuming an elementary knowledge of microprocessor programming. You will build and design your own domestic-use systems and peripherals. Self-test exercises included. $12.95.*

● **6502 ASSEMBLY LANGUAGE PROGRAMMING**—BK1176— by Lance A. Leventhal. This book provides comprehensive coverage of the 6502 microprocessor assembly language. Leventhal covers over 80 programming examples from simple memory load loops to complete design projects. Features include 6502 assembler conventions, input/output devices and interfacing methods, and programming the 6502 interrupt system. $16.99.*

● **6502 SOFTWARE GOURMET GUIDE AND COOKBOOK** — BK1055 — by Robert Findley. This book introduces the BASIC language programmer into the realm of machine-language programming. The description of the 6502 structure and instruction set, various routines, subroutines and programs are the ingredients in this cookbook. "Recipes" are included to help you put together exactly the programs to suit your taste. $12.95.*

● ***MICROCOMPUTING* CODING SHEETS** *Microcomputing*'s dozen or so programmers wouldn't try to work without these handy scratch pads, which help prevent the little errors that can cost hours and hours of programming time. Available for programming is Assembly/Machine Language (PD1001), which has columns for address, instruction (3 bytes), source code (label, op code, operand) and comments; and for BASIC (PD1002) which is 72 columns wide. 50 sheets to a pad. $2.39.*

—8080 / 8080A—

● **8080A/8085 Assembly Language Programming**—by Lance Leventhal—BK1004—Assembly language programming for the 8080A/8085 is explained with a description of the functions of assemblers and assembly instructions, and a discussion of basic software development concepts. Many fully debugged, practical programs are included as is a special section on structured programming. $15.99.*

● **8080 PROGRAMMING FOR LOGIC DESIGN**—BK1078—Ideal reference for an indepth understanding of the 8080 processor. Application-oriented and the 8080 is discussed in light of replacing conventional, hard-wired logic. Practical design considerations are provided for the implementation of an 8080-base control system. $9.50.*

● **8080 SOFTWARE GOURMET GUIDE AND COOKBOOK** — BK1102 — If yu have been spending too much time developing simple routines for your 8080, try this new book by Scelbi Computing and Robert Findley. Describes sorting, searching, and many other routines for the 8080 user. $12.95.*

—6800—

● **6800 PROGRAMMING FOR LOGIC DESIGN**—BK1077—Oriented toward the industrial user, this book describes the process by which conventional logic can be replaced by a 6800 microprocessor. Provides practical information that allows an experimenter to design a complete micro control system from the "ground up." $9.50.*

● **6800 SOFTWARE GOURMET GUIDE AND COOKBOOK** — BK1075 — Like its culinary cousin, *The 8080 Gourmet Guide*, this book by Scelbi Computing and Robert Findley describes sorting, searching and other routines — this time for the 6800 user. $12.95.*

—COOK BOOKS—

● **CMOS COOKBOOK** — BK1011 — by Don Lancaster. Details the application of CMOS, the low power logic family suitable for most applications presently dominated by TTL. Required reading for every serious digital experimenter! $10.50.*

● **TVT COOKBOOK** — BK1064 — by Don Lancaster. Describes the use of a standard television receiver as a microprocessor CRT terminal. Explains and describes character generation, cursor control and interface information in typical, easy-to-understand Lancaster style. $9.95.*

● **TTL COOKBOOK** — BK1063 — by Donald Lancaster. Explains what TTL is, how it works, and how to use it. Discusses practical applications, such as a digital counter and display system, events counter, electronic stopwatch, digital voltmeter and a digital tachometer. $9.50.*

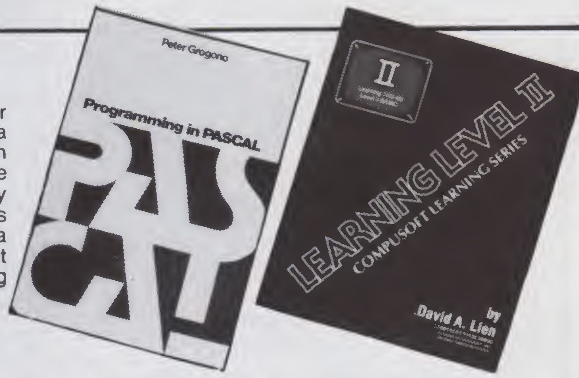
"PRICES SUBJECT TO CHANGE WITHOUT NOTICE"

*Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to Kilobaud *Microcomputing* Book Department • Peterborough NH 03458. Be sure to include check or detailed credit card information. No C.O.D. orders accepted. All orders add $1.00 handling. Please allow 4-6 weeks for delivery. Questions regarding your order? Please write to Customer Service at the above address.

FOR TOLL FREE ORDERING CALL 1-800-258-5473

kb microcomputing book nook

BASIC AND PASCAL



NEW REVISED EDITION

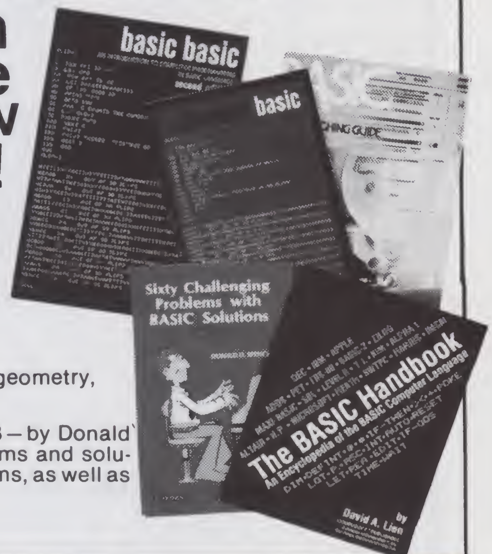
● **PROGRAMMING IN PASCAL**—BK1140—by Peter Grogono. The computer programming language PASCAL was the first language to embody in a coherent way the concepts of structured programming, which has been defined by Edsger Dijkstra and C.A.R. Hoare. As such, it is a landmark in the development of programming languages. PASCAL was developed by Niklaus Wirth in Zurich; it is derived from the language ALGOL 60 but is more powerful and easier to use. PASCAL is now widely accepted as a useful language that can be efficiently implemented, and as an excellent teaching tool. It does not assume knowledge of any other programming language; it is therefore suitable for an introductory course. $12.95.*

● **THE BASIC HANDBOOK**—BK1174—by David Lien. This book is unique. It is a virtual ENCYCLOPEDIA of BASIC. While not favoring one computer over another, it explains over 250 BASIC words, how to use them and alternate strategies. If a computer does not possess the capabilities of a needed or specified word, there are often ways to accomplish the same function by using another word or combination of words. That's where the HANDBOOK comes in. It helps you get the most from your computer, be it a "bottom-of-the-line" micro or an oversized monster. $14.95.*

● **LEARNING LEVEL II**—BK1175—by David Lien. Written especially for the TRS-80, this book concentrates on Level II BASIC, exploring every important BASIC language capability. Updates are included for those who have studied the Level I User's Manual. Sections include: how to use the Editor, dual cassette operation, printers and peripheral devices, and the conversion of Level I programs to Level II. $15.95.*

● **INTRODUCTION TO PASCAL**—BK1189—by Rodnay Zaks. A step-by-step introduction for anyone wanting to learn the language quickly and completely. Each concept is explained simply and in a logical order. All features of the language are presented in a clear, easy-to-understand format with exercises to test the reader at the end of each chapter. It describes both standard PASCAL and UCSD PASCAL, the most widely used dialect for small computers. No computer or programming experience is necessary. $12.95.*

new!



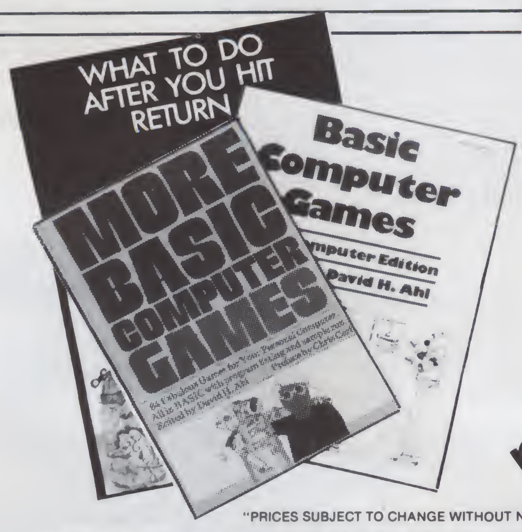
● **BASIC NEW 2ND EDITION**—BK1081—by Bob Albrecht. Self-teaching guide to the computer language you will need to know for use with your microcomputer. This is one of the easiest ways to learn computer programming. $6.95.*

● **BASIC BASIC (2ND EDITION)**—BK1026—by James S. Coan. This is a textbook which incorporates the learning of computer programming using the BASIC language with the teaching of mathematics. Over 100 sample programs illustrate the techniques of the BASIC language and every section is followed by practical problems. This second edition covers character string handling and the use of data files. $10.50.*

● **ADVANCED BASIC**—BK1000—Applications, including strings and files, coordinate geometry, area, sequences and series, simulation, graphing and games. $10.75*.

● **SIXTY CHALLENGING PROBLEMS WITH BASIC SOLUTIONS (2nd Edition)**—BK1073—by Donald Spencer, provides the serious student of BASIC programming with interesting problems and solutions. No knowledge of math above algebra required. Includes a number of game programs, as well as programs for financial interest, conversions and numeric manipulations. $6.95.*

GAMES



● **WHAT TO DO AFTER YOU HIT RETURN**—BK1071—PCC's first book of computer games . . . 48 different computer games you can play in BASIC . . . programs, descriptions, many illustrations. Lunar Landing, Hammurabi, King, Civel 2, Qubic 5, Taxman, Star Trek, Crash, Market, etc. $10.95.*

● **BASIC COMPUTER GAMES**—BK1074—Okay, so once you get your computer and are running in BASIC, then what? Then you need some programs in BASIC, that's what. This book has 101 games for you from very simple to real buggers. You get the games, a description of the games, the listing to put in your computer and a sample run to show you how they work. Fun. Any one game will be worth more than the price of the book for the fun you and your family will have with it. $7.50.*

new!

● **MORE BASIC COMPUTER GAMES**—BK1182—edited by David H. Ahl. More fun in BASIC! 84 new games from the people who brought you *BASIC Computer Games*. Includes such favorites as Minotaur (battle the mythical beast) and Eliza (unload your troubles on the doctor at bargain rates). Complete with game description, listing and sample run. $7.50.*

"PRICES SUBJECT TO CHANGE WITHOUT NOTICE"

*Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to Kilobaud *Microcomputing* Book Department • Peterborough NH 03458. Be sure to include check or detailed credit card information. No C.O.D. orders accepted. All orders add $1.00 handling. Please allow 4-6 weeks for delivery. Questions regarding your order? Please write to Customer Service at the above address.

FOR TOLL FREE ORDERING CALL 1-800-258-5473

the CompuPro Systems Approach:

High Performance, High Throughput.

Unlike "all-in-one" computers, **CompuPro's** modular S-100 systems are amazingly flexible machines that are ideal for high level industrial, commercial, and scientific applications. Full conformance to all IEEE 696/S-100 specifications ensures well integrated systems performance, as well as freedom from obsolescence in the years to come.

All **CompuPro** products meet the most demanding mechanical and electrical standards, accept the highest possible clock speeds for maximum throughput, and are backed with one of the best - if not the best - warranties in the business (1 year limited warranty on all products, 2 year limited warranty with board exchange program for boards qualified under the Certified System Component program).

When you're looking for a computer, there are lots of choices. But when you need a precision machine that is built for the future as well as the present, the choice narrows down to the most experienced name in the S-100 business: **CompuPro.**

NEW! COMPUTER ENCLOSURE 2

Introductory price: $795 (specify rack mount or desk top version)

Now it's easy to move up to an expandable S-100 system. . .**COMPUTER ENCLOSURE 2** is ready to accept boards the minute it's unpacked. Fused, constant voltage power supply provides +8A at 25 Amps (!), +16V at 3 Amps, and -16V at 3 Amps. Also includes 20 slot shielded/active terminated motherboard, dual AC outlets on rear, heavy-duty line filter, circuit breaker, quiet ventilation fan, reset switch, and black anodized front panel (desktop version includes textured vinyl painted cover). Rack mount version includes slides for easy pull-out from rack frame.

Also available: COMPUTER ENCLOSURE 1. Same as above, but less power supply and motherboard. **$289** desktop, **$329** rack mount.

SYSTEM SUPPORT 1 MULTIFUNCTION BOARD

$295 Unkit, **$395** A/T, **$495** CSC

This multi-purpose S-100 board provides your computer with the most needed system support functions - at less cost than buying numerous single function boards. Includes sockets for 4K of extended address EPROM or RAM (2716 pinout), 1 socket with battery backup; crystal controlled month/day/year/time clock with BCD outputs; optional high speed math processor (9511 or 9512); full RS-232 serial port; three 16 bit interval timers (cascade or use independently); two interrupt controllers service 15 levels of interrupts; power fail indicator with provision to switch CMOS memory to battery backup; and comprehensive owner's manual with numerous software examples. Conforms fully to all IEEE 696/S-100 standards.

Want to make your S-100 system more versatile? **System Support 1** is the answer. (Add $195 to the above prices for the optional 9512 math processor.)

NEW LOWER PRICE ON 16K MEMORY EXPANSION-8 RAMS/$35

Just what you would expect from the memory leader: top quality, low power, high speed (200 ns) 16K dynamic RAMs, backed up with a 1 year limited warranty. Expand memory in TRS-80* -I, -II and -III computers (Color model too) as well as machines made by Apple, Exidy, Heath H89, newer PETs, etc. Add $3 for two dip shunts plus TRS-80* conversion instructions. **Limited quantity.**

S-100 HIGH PERFORMANCE MOTHERBOARDS

Actively terminated, fully shielded motherboards handle clock speeds up to 10 MHz. **Unkits** have edge connectors and termination resistors pre-soldered in place for easy assembly.

20 slot motherboard with edge connectors - Unkit **$174**, A/T **$214**
12 slot motherboard with edge connectors - Unkit **$129**, A/T **$169**
6 slot motherboard with edge connectors - Unkit **$89**, A/T **$129**

OTHER S-100 BUS PRODUCTS

Active Terminator Board	**$34.50** Kit
Memory Manager Board	**$59** Unkit, **$85** A/T, **$100** CSC
Mullen Extender Board	**$59** Kit
Mullen Relay/Opto-Isolator Control Board	**$129** Kit, **$179** A/T
Spectrum color graphics board	**$299** Unkit, **$399** A/T, **$499** CSC
2708 EPROM Board (2708s not included)	**$85** Unkit, **$135** A/T, **$195** CSC
Interfacer 1 (dual RS-232 serial ports)	**$199** Unkit, **$249** A/T, **$324** CSC
Interfacer 2 (3 parallel + 1 serial port)	**$199** Unkit, **$249** A/T, **$324** CSC

LEGAL CORNER Z80A is a registered trademark of Zilog. TRS-80 is a trademark of the Tandy Corporation; PASCAL M is a trademark of Sorcim. CP/M is a registered trademark of Digital Research

HIGH SPEED S-100 CPU BOARDS

8 BIT CPU Z

Like many others, we claim full conformance to IEEE 696/S-100 specifications; unlike many others, we'll send you the timing specs to prove it. **CPU Z** includes all standard Z-80A* features along with power on jump, on-board fully maskable interrupts for interrupt-driven systems, selectable automatic wait state insertion, provision for adding up to 8K of on-board EPROM, and 16/24 bit extended addressing. Works with 6 MHz CPUs; supplied with 4 MHz CPU. **$225** Unkit, **$295** A/T, **$395** CSC.

16/8 BIT CPU 8085/88

The only board that bridges the 8 bit world of the present and the 16 bit world of the future. By using an 8088 CPU (for 16 bit power with a standard 8 bit bus) in conjunction with an 8 bit 8085, **CPU 8085/88** is downward compatible with 8080/8085 software, upward compatible with 8086/88 software (as well a Intel's coming P-Series), designed for professional-level high speed applications, and capable of accessing 16 megabytes of memory. . .while conforming fully to all IEEE 696/S-100 standards (timing specs available on request).

Looking for a powerful 8 bit CPU board? Looking for a powerful 16 bit CPU board? Then look at **CPU 8085/88**, the best of both worlds.

Prices: $295 Unkit, **$425** A/T (both operate at 5 MHz); **$525** CSC (with 5 MHz 8085, 6 MHz 8088). Owner's manual available separately for **$5.**

8 BIT CPU 8085

This is a single 8 bit processor version of the above board, and may be easily upgraded to full 16 bit operation at a later date. **$235** Unkit, **$325** A/T, **$425** CSC.

SOFTWARE

8088/8086 MONITOR-DEBUGGER: $35

Supplied on single sided, single density, soft-sector 8" disc. CP/M* compatible. Great development tool; mnemonics used in debug conform as closely as possible to current CP/M* DDT mnemonics.

PASCAL/M* FROM SORCIM: $175 COMPLETE

PASCAL - easy to learn, easy to apply - can give a microcomputer with CP/M* more power than many minis. We supply a totally standard **Wirth PASCAL/M* 8" diskette** and comprehensive manual. Specify Z-80* or 8080/8085 version.

S-100 MEMORIES

CompuPro memories feature fully static design to eliminate dynamic timing problems, full conformance to all IEEE 696/S-100 specifications, high speed operation (4/5 MHz Unkit, 10 MHz A/T and CSC), low power consumption, extensive bypassing, and careful thermal design.

	Unkit	A/T	CSC
8K RAM 2A	**$159**	**$189**	**$239**
16K RAM 14 (extended addressing)	**$279**	**$349**	**$429**
16K RAM 20-16 (extended addressing and bank select)	**$319**	**$399**	**$479**
24K RAM 20-24 (extended addressing and bank select)	**$429**	**$539**	**$629**
32K RAM 20-32 (extended addressing and bank select)	**$559**	**$699**	**$799**
128K RAM 21-128 (extended addressing)	n/a	n/a	**$2795**

Most **CompuPro** products are available in **Unkit** form, **Assembled/Tested**, or qualified under the high-reliability **Certified System Component (CSC)** program (200 hour burn-in, more). **Note: Unkits** are not intended for novices, as de-bugging may be required due to problems such as IC infant mortality. Factory service is available for **Unkits** at a flat service charge.

TERMS: Prices shown do not include dealer installation and support services. Cal res add tax. Allow at least 5% shipping; excess refunded. Orders under $15 add $2 handling. VISA and Mastercard orders ($25 min) call our 24 hour order desk at **(415) 562-0636.** Include street address for UPS delivery. Prices are subject to change without notice.

FREE CATALOG: Want more information? Then send for our free catalog. For fast 1st class delivery, add 41 cents in stamps; foreign orders add $2 (refundable with order).

CompuPro™ division of GODBOUT ELECTRONICS

OAKLAND AIRPORT, CA 94614 (415) 562-0636

42

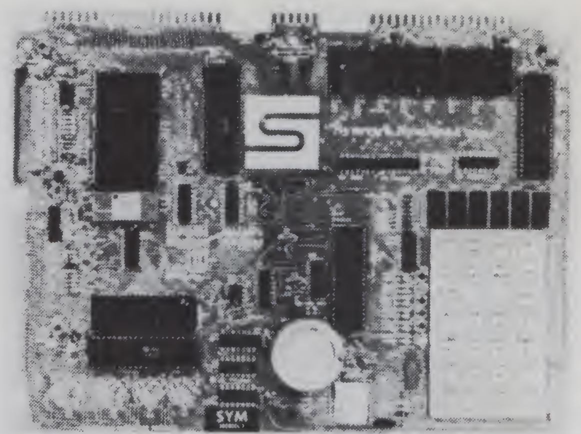
COMPUPRO PRODUCTS ARE AVAILABLE AT COMPUTER STORES WORLD-WIDE. . .CALL (415) 563-0636 FOR THE STORE NEAREST YOU.

Reader Service—see page 178

✓Reader Service—see page 178

SYM-1, 6502-BASED MICROCOMPUTER

- FULLY-ASSEMBLED AND COMPLETELY INTEGRATED SYSTEM that's ready-to-use
- ALL LSI IC'S ARE IN SOCKETS
- 28 DOUBLE-FUNCTION KEYPAD INCLUDING UP TO 24 "SPECIAL" FUNCTIONS
- EASY-TO-VIEW 6-DIGIT HEX LED DISPLAY
- KIM-1* HARDWARE COMPATIBILITY
  The powerful 6502 8-Bit MICROPROCESSOR whose advanced architectural features have made it one of the largest selling "micros" on the market today.
- THREE ON-BOARD PROGRAMMABLE INTERVAL TIMERS available to the user, expandable to five on-board.
- 4K BYTE ROM RESIDENT MONITOR and Operating Programs.
- Single 5 Volt power supply is all that is required.
- 1K BYTES OF 2114 STATIC RAM onboard with sockets provided for immediate expansion to 4K bytes onboard, with total memory expansion to 65, 536 bytes.
- USER PROM/ROM: The system is equipped with 3 PROM/ROM expansion sockets for 2316/2332 ROMs or 2716 EPROMs
- ENHANCED SOFTWARE with simplified user interface
- STANDARD INTERFACES INCLUDE:
  - —Audio Cassette Recorder Interface with Remote Control (Two modes: 135 Baud KIM-1* compatible, Hi-Speed 1500 Baud)
  - —Full duplex 20mA Teletype Interface
  - —System Expansion Bus Interface
  - —TV Controller Board Interface
  - —CRT Compatible Interface (RS-232)
- APPLICATION PORT: 15 Bi-directional TTL Lines for user applications with expansion capability for added lines
- EXPANSION PORT FOR ADD-ON MODULES (51 I/O Lines included in the basic system)
- SEPARATE POWER SUPPLY connector for easy disconnect of the d-c power
- AUDIBLE RESPONSE KEYPAD



Synertek has enhanced KIM-1* software as well as the hardware. The software has simplified the user interface. The basic SYM-1 system is programmed in machine language. Monitor status is easily accessible, and the monitor gives the keypad user the same full functional capability of the TTY user. The SYM-1 has everything the KIM-1* has to offer, plus so much more that we cannot begin to tell you here. So, if you want to know more, the SYM-1 User Manual is available, separately

SYM-1 Complete w/manuals	**$229.00**
SYM-1 User Manual Only	**$7.00**
SYM-1 Expansion	**$60.00**

Expansion includes 3K of 2114 RAM chips and 1-6522 I/O chip.

SYM-1 Manuals: The well organized documentation package is complete and easy-to-understand.

SYM-1 CAN GROW AS YOU GROW. It's the system to BUILD-ON. Expansion features that are available:

BAS-1 8K Basic ROM (Microsoft Basic)	**$89.00**
KTM-2 (Complete terminal less monitor)	**$319.00**

QUALITY EXPANSION BOARDS DESIGNED SPECIFICALLY FOR KIM-1, SYM-1 & AIM 65

These boards are set up for use with a regulated power supply such as the one below, but, provisions have been made so that you can add onboard regulators for use with an unregulated power supply. But, because of unreliability, we do not recommend the use of onboard regulators. All I.C.'s are socketed for ease of maintenance. *All boards carry full 90-day warranty.*

All products that we manufacture are designed to meet or exceed industrial standards. All components are first qualtiy and meet full manufacturer's specifications. All this and an extended burn-in is done to reduce the normal percentage of field failures by up to 75%. To you, this means the chance of inconvenience and lost time due to a failure is very rare; but, if it should happen, we guarantee a turn-around time of less than forty-eight hours for repair.

Our money back guarantee: If, for any reason you wish to return any board that you have purchased directly from us within ten (10) days after receipt, complete, in original condition, and in original shipping carton; we will give you a complete credit or refund less a $10.00 restocking charge per board.

VAK-1 8-SLOT MOTHERBOARD

This motherboard uses the KIM-4* bus structure. It provides eight (8) expansion board sockets with rigid card cage. Separate jacks for audio cassette, TTY and power supply are provided. Fully buffered bus.

VAK-1 Motherboard **$139.00**

VAK-2/4 16K STATIC RAM BOARD

This board using 2114 RAMs is configured in two (2) separately addressable 8K blocks with individual write-protect switches.

VAK-2 16K RAM Board with only 8K of RAM (½ populated)	**$239.00**
VAK-3 Complete set of chips to expand above board to 16K	**$125.00**
VAK-4 Fully populated 16K RAM	**$325.00**

VAK-5 2708 EPROM PROGRAMMER

This board requires a +5 VDC and ±12 VDC, but has a DC to DC multiplyer so there is no need for an additional power supply. All software is resident in on-board ROM, and has a zero-insertion socket.

VAK-5 2708 EPROM Programmer **$249.00**

VAK-6 EPROM BOARD

This board will hold 8K of 2708 or 2758, or 16K of 2716 or 2516 EPROMs. *EPROMs not included.*

VAK-6 EPROM Board **$119.00**

VAK-7 COMPLETE FLOPPY-DISK SYSTEM $1299.00

See May Kilobaud for details

VAK-8 PROTYPING BOARD

This board allows you to create your own interfaces to plug into the motherboard. Etched circuitry is provided for regulators, address and data bus drivers; with a large area for either wire-wrapped or soldered IC circuitry.

VAK-8 Protyping Board **$39.00**

POWER SUPPLIES

ALL POWER SUPPLIES are totally enclosed with grounded enclosures for safety, AC power cord, and carry a full 2-year warranty.

FULL SYSTEM POWER SUPPLY

This power supply will handle a microcomputer and up to 65K of our VAK-4 RAM. ADDITIONAL FEATURES ARE: Over voltage Protection on 5 volts, fused, AC on/off switch. Equivalent to units selling for $225.00 or more.

Provides +5 VDC @ 10 Amps & ±12 VDC @ 1 Amp

VAK-EPS Power Supply **$119.00**

VAK-EPS/AIM—same as VAK-EPS but w/additional 24 volt unregulated (specifically for AIM 65) **$149.00**

KIM-1* Custom P.S. provides 5 VDC @ 1.2 Amps and +12 VDC @ .1 Amps

KCP-1 Power Supply **$39.00**

*KIM is a product of MOS Technology

Add $2.50 for shipping & handling for all <u>except</u> AIM 65.

RNB ENTERPRISES

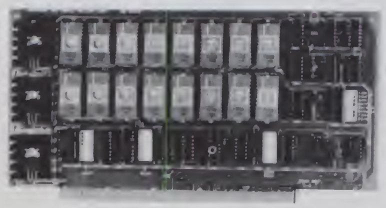
INCORPORATED

✓52

WE'VE EXPANDED AGAIN!
NOTE OUR NEW ADDRESS.
2951 W. Fairmount Avenue
Phoenix AZ 85017
(602)265-7564

DIGITAL RESEARCH COMPUTERS
(214) 271-3538

32K S-100 EPROM CARD
NEW!



$74.95
KIT

USES 2716's

Blank PC Board - $34

ASSEMBLED & TESTED ADD $30

SPECIAL: 2716 EPROM's (450 NS) Are $11.95 EA. With Above Kit.

KIT FEATURES:
1. Uses +5V only 2716 (2Kx8) EPROM's.
2. Allows up to 32K of software on line!
3. IEEE S-100 Compatible.
4. Addressable as two independent 16K blocks.
5. Cromemco extended or Northstar bank select.
6. On board wait state circuitry if needed.
7. Any or all EPROM locations can be disabled.
8. Double sided PC board, solder-masked, silk-screened.
9. Gold plated contact fingers.
10. Unselected EPROM's automatically powered down for low power.
11. Fully buffered and bypassed.
12. Easy and quick to assemble.

32K SS-50 RAM

$379.00 KIT

NEW!

For 2MHZ Add $10

Blank PC Board $50

For SWTPC 6800 - 6809 Buss

Support IC's and Caps $19.95

Complete Socket Set $21.00

Fully Assembled, Tested, Burned In Add $30

At Last! An affordable 32K Static RAM with full 6809 Capability.

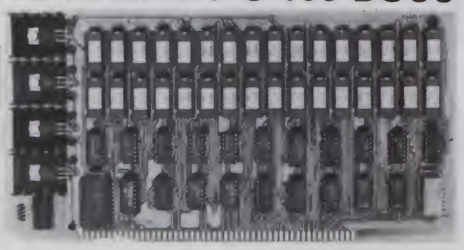
FEATURES:
1. Uses proven low power 2114 Static RAMS.
2. Supports SS50C - EXTENDED ADDRESSING.
3. All parts and sockets included.
4. Dip Switch address select as a 32K block.
5. Extended addressing can be disabled.
6. Works with all existing 6800 SS50 systems.
7. Fully bypassed. PC Board is double sided, plated thru, with silk screen.

16K STATIC RAM KIT-S 100 BUSS

PRICE CUT!

$199.95 KIT

FOR 4MHZ ADD $10



KIT FEATURES:
1. Addressable as four separate 4K Blocks.
2. ON BOARD BANK SELECT circuitry. (Cromemco Standard!). Allows up to 512K on line!
3. Uses 2114 (450NS) 4K Static Rams.
4. ON BOARD SELECTABLE WAIT STATES.
5. Double sided PC Board, with solder mask and silk screened layout. Gold plated contact fingers.
6. All address and data lines fully buffered.
7. Kit includes ALL parts and sockets.
8. PHANTOM is jumpered to PIN 67.
9. LOW POWER: under 1.5 amps TYPICAL from the +8 Volt Buss
10. Blank PC Board can be populated as any multiple of 4K.

BLANK PC BOARD W/DATA-**$33**
LOW PROFILE SOCKET SET-**$12**
SUPPORT IC'S & CAPS-**$19.95**
ASSEMBLED & TESTED-**ADD $35**

OUR #1 SELLING RAM BOARD!

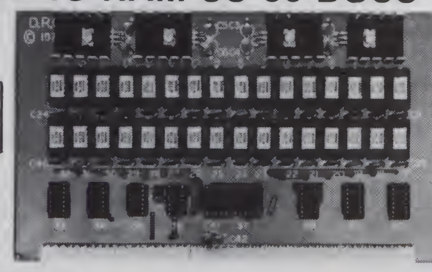
16K STATIC RAM SS-50 BUSS

PRICE CUT!

$195 KIT

FULLY STATIC!

FOR 2MHZ ADD $10



FOR SWTPC 6800 BUSS!

ASSEMBLED AND TESTED - $35

KIT FEATURES:
1. Addressable on 16K Boundaries
2. Uses 2114 Static Ram
3. Fully Bypassed
4. Double sided PC Board. Solder mask and silk screened layout.
5. All Parts and Sockets included
6. Low Power: Under 1.5 Amps Typical

BLANK PC BOARD—**$35** COMPLETE SOCKET SET—**$12**
SUPPORT IC'S AND CAPS—**$19.95**

NEW! STEREO! S-100 SOUND COMPUTER BOARD NEW!

At last, an S-100 Board that unleashes the full power of two unbelievable General Instruments AY3-8910 NMOS computer sound IC's. Allows you under total computer control to generate an infinite number of special sound effects for games or any other program. Sounds can be called in BASIC, ASSEMBLY LANGUAGE, etc.

COMPLETE KIT! $84.95 (WITH DATA MANUAL)

BLANK PC BOARD W/DATA $31

KIT FEATURES:
* TWO GI SOUND COMPUTER IC'S.
* FOUR PARALLEL I/O PORTS ON BOARD.
* USES ON BOARD AUDIO AMPS OR YOUR STEREO.
* ON BOARD PROTO TYPING AREA.
* ALL SOCKETS, PARTS AND HARDWARE ARE INCLUDED.
* PC BOARD IS SOLDERMASKED, SILK SCREENED, WITH GOLD CONTACTS.
* EASY, QUICK, AND FUN TO BUILD. WITH FULL INSTRUCTIONS.
* USES PROGRAMMED I/O FOR MAXIMUM SYSTEM FLEXIBILITY.

Both Basic and Assembly Language Programming examples are included.

SOFTWARE:
SCL™ is now available! Our Sound Command Language makes writing Sound Effects programs a SNAP! SCL™ also includes routines for Register-Examine-Modify, Memory-Examine-Modify, and Play-Memory. SCL™ is available on CP/M* compatible diskette or 2708 or 2716. Diskette - **$24.95** 2708 - **$19.95** 2716 - **$29.95.** Diskette includes the source. EPROM'S are ORG at E000H.

4K DYNAMIC RAM BLOWOUT! SAME AS INTEL 2107B! 4K RAMS AT AN UNBELIEVABLE 50¢ EACH!!!

Prime, new, National Semi., 1979 date coded, full spec. parts. N.S. #MM5280-5N. Same as INTEL 2107B-4, T.I. TMS4060, NEC uPD411, etc. We bought a HUGE QTY. from a West Coast Distributor at truly DISTRESS PRICES! One of the most popular and reliable RAM's ever made. These parts have been used by almost all Major Computer Main Frame Mfg. the world over! Arranged as 4K x 1, 270 NS Access Time, 22 Pin Dip. These units DO NOT use multiplexed addressing, thus making REFRESH and other timing very simple. See INTEL MEMORY DESIGN HANDBOOK for full application notes. The NAT. SEMI. MEMORY DATA BOOK is available at most Radio Shack Stores. Prime units in original factory tubes!

(With Pin Out Data)

#5280-5N 4096 BITS x 1 270 NS ACCESS
8 FOR $4.95 32 FOR $16
FACTORY CASE (450 PCS) — $180
Sockets Special: 22 Pin Low Profile (With Purchase of 5280's) 8 FOR $1.

COMPUTER PARTS SPECIALS

74LS175 - .99	8035 Intel Single Chip CPU **6.95**
74LS240 - **1.19**	Signetics 2901 4 Bit Slice - **6.95**
74LS241 - **1.19**	AMD 2903 4 Bit Super Slice - **12.50**
74LS244 - **1.19**	AMD 29705 Dual Port RAM - **8.95**
74LS373 - **1.29**	Intel 2716-1 (350 NS) - **12.95**

NEW! G.I. COMPUTER SOUND CHIP

AY3-8910. As featured in July, 1979 BYTE! A fantastically powerful Sound & Music Generator. Perfect for use with any 8 Bit Microprocessor. Contains: 3 Tone Channels, Noise Generator, 3 Channels of Amplitude Control, 16 bit Envelope Period Control, 2-8 Bit Parallel I/O, 3 D to A Converters, plus much more! All in one 40 Pin DIP. Super easy interface to the S-100 or other busses. **$11.95 PRICE CUT!**

SPECIAL OFFER: ~~$14.95~~ each Add $3 for 60 page Data Manual.

Digital Research Computers
(OF TEXAS)

P.O. BOX 401565 • GARLAND, TEXAS 75040 • (214) 271-3538

TERMS: Add $1.50 postage. We pay balance. Orders under $15 add 75¢ handling. No C.O.D. We accept Visa and MasterCharge. Tex. Res. add 5% Tax. Foreign orders (except Canada) add 20% P & H. Orders over $50, add 85¢ for insurance.

ALL SALES ARE MADE SUBJECT TO THE TERMS OF OUR 90 DAY LIMITED WARRANTY. A COPY OF THIS WARRANTY IS AVAILABLE FREE, ON REQUEST.

*TRADEMARK OF DIGITAL RESEARCH.

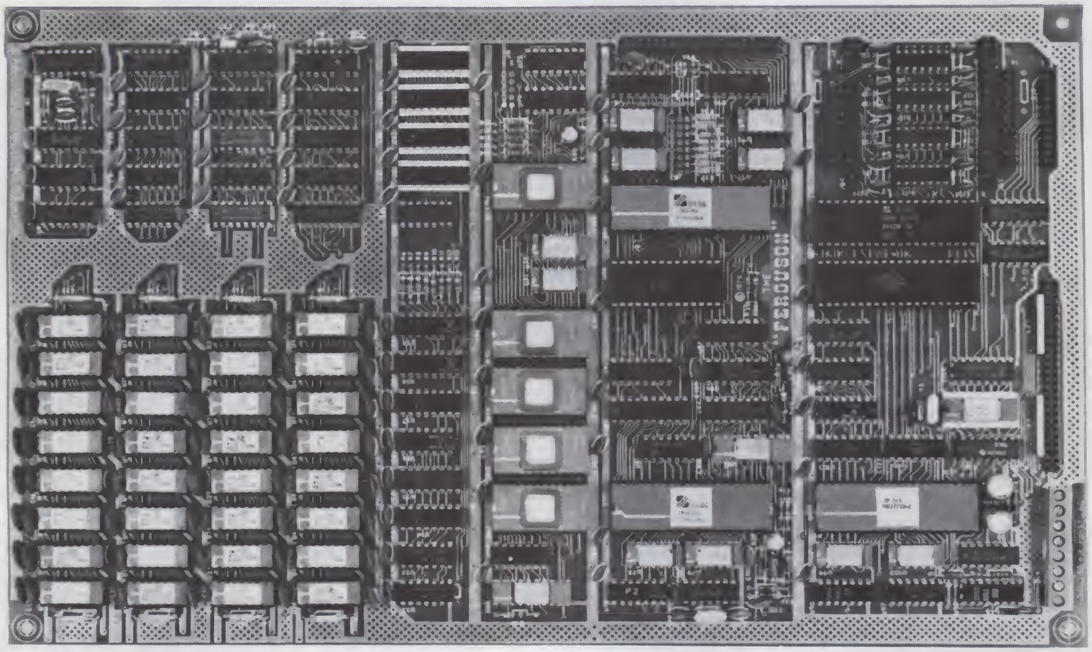
WE ARE NOT ASSOCIATED WITH DIGITAL RESEARCH OF CALIFORNIA, THE SUPPLIERS OF CPM SOFTWARE.

"THE BIG BOARD"

OEM - INDUSTRIAL - BUSINESS - SCIENTIFIC

NEW! SINGLE BOARD COMPUTER KIT! NEW!

Z-80 CPU! 64K RAM!



PARTIAL KIT
For All Sockets Installed And Soldered Add $50.

THE FERGUSON PROJECT: Three years in the works, and maybe too good to be true. A tribute to hard headed, no compromise, high performance, American engineering! The Big Board gives you all the most needed computing features on one board at a very reasonable cost. The Big Board was designed from scratch to run the latest version of CP/M*. Just imagine all the off-the-shelf software that can be run on the Big Board without any modifications needed! Take a Big Board, add a couple of 8 inch disc drives, power supply, an enclosure, C.R.T., and you have a total Business System for about 1/3 the cost you might expect to pay.

$649.00** (64K KIT BASIC I/O)

SIZE: 8½ x 13¾ IN.
SAME AS AN 8 IN. DRIVE.
REQUIRES: +5V @ 3 AMPS
+ - 12V @ .5 AMPS.

FULLY SOCKETED!

FEATURES: (Remember, all this on one board!)

64K RAM

Uses industry standard 4116 RAM'S. All 64K is available to the user, our VIDEO and EPROM sections do not make holes in system RAM. Also, very special care was taken in the RAM array PC layout to eliminate potential noise and glitches.

Z-80 CPU

Running at 2.5 MHZ. Handles all 4116 RAM refresh and supports Mode 2 INTERUPTS. Fully buffered and runs 8080 software.

SERIAL I/O (OPTIONAL)

Full 2 channels using the Z80 SIO and the SMC 8116 Baud Rate Generator. FULL RS232! For synchronous or asynchronous communication. In synchronous mode, the clocks can be transmitted or received by a modem. Both channels can be set up for either data-communication or data-terminals. Supports mode 2 Int. Price for all parts and connectors: $85.

BASIC I/O

Consists of a separate parallel port (Z80 PIO) for use with an ASCII encoded keyboard for input. Output would be on the 80 x 24 Video Display.

24 x 80 CHARACTER VIDEO

With a crisp, flicker-free display that looks extremely sharp even on small monitors. Hardware scroll and full cursor control. Composite video or split video and sync. Character set is supplied on a 2716 style ROM, making customized fonts easy. Sync pulses can be any desired length or polarity. Video may be inverted or true. 5 x 7 Matrix - Upper & Lower Case

FLOPPY DISC CONTROLLER

Uses WD1771 controller chip with a TTL Data Separator for enhanced reliability. IBM 3740 compatible. Supports up to four 8 inch disc drives. Directly compatible with standard Shugart drives such as the SA800 or SA801. Drives can be configured for remote AC off-on. Runs CP/M* 2.2.

TWO PORT PARALLEL I/O (OPTIONAL)

Uses Z-80 PIO. Full 16 bits, fully buffered, bi-directional. User selectable hand shake polarity. Set of all parts and connectors for parallel I/O: $29.95

REAL TIME CLOCK (OPTIONAL)

Uses Z-80 CTC. Can be configured as a Counter on Real Time Clock. Set of all parts: $14.95

SYSTEM COMPARISON

64K RAM KIT	$370.00
80 x 24 Video Kit	365.00
Floppy Disk Controller Kit	235.00
Z-80 CPU Kit	185.95
SER & PAR. I/O	129.95
S-100 Mother Board	45.00
SUB TOTAL	$1330.90

Talk about bangs per buck! The prices shown for S100 kits were taken from the July 1980 BYTE. This will give some basis for comparison between the Big Board and a similar system implementation on the S100 Buss.

CP/M* 2.2 FOR BIG BOARD

The popular CP/M* D.O.S. modified by MICRONIX SYSTEMS to run on Big Board is available for $150.00.

PC BOARD

Blank PC Board with Rom Set and Full Documentation.
$195.00

PFM 3.0 2K SYSTEM MONITOR

The real power of the Big Board lies in its PFM 3.0 on board monitor. PFM commands include: Dump Memory, Boot CP/M*, Copy, Examine, Fill Memory, Test Memory, Go To, Read and Write I/O Ports, Disc Read (Drive, Track, Sector), and Search. PFM occupies one of the four 2716 EPROM locations provided.

Z-80 is a Trademark of Zilog.

Digital Research Computers
(OF TEXAS)
P.O. BOX 401565 • GARLAND, TEXAS 75040 • (214) 271-3538

TERMS: Initial shipments will be made approximately 3 to 5 weeks after we receive your order. VISA, MC, cash accepted. We will accept COD's (for the Big Board only) with a $75 deposit. Balance UPS COD. The $75 deposit assures your place in line for the initial production run of Big Board.

*TRADEMARK OF DIGITAL RESEARCH. NOT ASSOCIATED WITH DIGITAL RESEARCH OF CALIFORNIA, THE ORIGINATORS OF CPM SOFTWARE
**1 TO 4 PIECE DOMESTIC USA PRICE.

SALE SALE SALE

Disk Drives

JADE's new dual disk sub-assemblies include: Handsome metal cabinet with proportionally balanced air flow system, rugged dual drive power supply, cooling fan, cable kit, lighted power switch, approved fuse assembly, line cord, Never-Mar rubber feet, and all necessary hardware to mount 2-8" disk drives - it's all American made, guaranteed for six months, and it's in stock!

Dual 8" Sub-Assembly Cabinet

END-000421 *Cabinet kit* **$225.00**
END-000420 *Bare cabinet* **$59.95**

Single sided, double density disk drive sub-system

END-000423 *Kit w/2 8" drives* **$975.00**
END-000424 *A & T w/2 8" drives* **$1195.00**

Double sided, double density disk drive sub-system

END-000426 *kit w/2 8" drives* **$1495.00**
END-000427 *A & T w/2 8" drives* **$1695.00**

JADE DISK PACKAGE

Double density controller, two 8" double density floppy disk drives, CP/M 2.2 (configured for controller), hardware and software manuals, boot PROM, cabinet, power supply, fan, & cables

Special package price **$1395.00**

8" Disk Drive Sale

Highly reliable double density floppy disk drives

Shugart 801R single sided, double density

MSF-10801R *SA-801R* **$425.00**
Special Sale Price .... ***2 for $790.00***

Siemens FDD100-8D2 single sided, double density

MSF-201120 *6 mo warranty* **$385.00**
Special sale price .... ***2 for $750.00***

Real Double-Sided Drives

8" Double-Sided Double-Density Sale

* ***Shugart SA-851R double-sided, double-density*** *
* ***only $625.00 ea*** .... ***2 for $1190.00*** *

MFE M701 8" double-sided, double-density drives
only $525 ea .... ***2 for $1040.00***

Qume Data Track 8 double-sided, double-density drives
only $575.00 .... ***2 for $1100.00***

Printers

CENTRONICS 737-1

9 x N dot matrix, letter quality, proportional spacing

PRM-15737 *Parallel* **$795.00**
With interface for Apple **$895.00**

MX-80 - Epson

132 column, 9 x 9 dot matrix, multiple fonts

PRM-27080 *Save $100.00* **$545.00**
Interface for Apple **$110.00**

SPINWRITER - NEC

65 cps, bi-directional, letter quality printer with deluxe tractor mechanism, both parallel and serial interfaces on-board, 16K buffer, ribbon, print thimble, graphics, micro space justification, data cable, and self test diagnostic ROM.

PRD-55511 *without 16K buffer* ... **$2795.00**
PRD-55512 *with 16K buffer* **$2895.00**

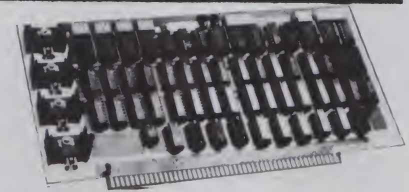
S-100 Systems

S-100 SYSTEM - Calif Computer Sys

Complete S-100 system including 12 slot mainframe, 4 MHz Z-80 CPU, 64K RAM memory, double density disk controller, RS-232 cable, 8" & 5¼" disk drive cables, CP/M 2.2, manuals, auto boot ROM, completely assembled & tested.

2210A *Integrated & tested* **$1995.00**
2210B *Not integrated* **$1795.00**

S-100 Memory



64K RAM - Calif Computer Sys

4 MHz bank port / bank byte selectable, extended addressing, 16K bank selectable, PHANTOM line allows memory overlay, 8080 / Z-80 / front panel compatible.

MEM-64565A *A & T* **$449.95**

MEMORY BANK - Jade

4 MHz, IEEE S-100, bank selectable, 8 or 16 bit

MEM-99730B *Bare board* **$55.00**
MEM-99730K *Kit, no RAM* **$219.95**
MEM-16730K *16K kit* **$249.95**
MEM-32731K *32K kit* **$289.95**
MEM-48732K *48K kit* **$324.95**
MEM-64733K *64K kit* **$359.95**
Assembled & tested .... add **$50.00**

EXPANDORAM II - S D Systems

4 MHz RAM board expandable from 16K to 256K

MEM-16630K *16K kit* **$275.95**
MEM-32631K *32K kit* **$295.95**
MEM-48632K *48K kit* **$315.95**
MEM-64633K *64K kit* **$335.95**
Assembled & tested .... add **$50.00**

32K STATIC RAM - Jade

2 or 4 MHz expandable static RAM board uses 2114L's

MEM-16151K *16K 4 MHz kit* **$169.95**
MEM-32151K *32K 4 MHz kit* **$299.95**
Assembled & tested .... add **$50.00**

16K STATIC RAM - Cal Comp Sys

2 or 4 MHz 16K static RAM board, IEEE S-100, bank selectable, Phantom capability, addressable in 4K blocks

MEM-16160A *16K 2 MHz A & T* ... **$286.95**
MEM-16162A *16K 4 MHz A & T* ... **$289.95**
MEM-16160B *Bare board* **$50.00**

PB-1 - S.S.M.

2708, 2716 EPROM board with built-in programmer

MEM-99510K *Kit* **$154.95**
MEM-99510A *A & T* **$229.95**

PROM-100 - SD Systems

2708, 2716, 2732, 2758, & 2516 EPROM programmer

MEM-99520K *Kit* **$219.95**
MEM-99520A *Jade A & T* **$269.95**

S-100 Video

VB-3 - S.S.M.

80 characters x 24 lines expandable to 80 x 48 for a full page of text, upper & lower case, 256 user defined symbols, 160 x 192 graphics matrix, memory mapped, has key board input.

IOV-1095K *4 MHz kit* **$375.00**
IOV-1095A *4 MHz A & T* **$450.00**
IOV-1096K *80 x 48 upgrade* **$39.95**

VIDEO BOARD - Jade

64 characters x 16 lines, 7 x 9 dot matrix, full upper/lower case ASCII character set, numbers, symbols, and greek letters, normal/reverse/blinking video, S-100.

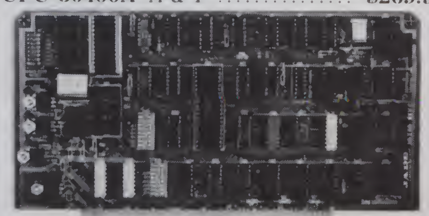
IOV-1050K *Kit* **$99.95**
IOV-1050A *A & T* **$125.00**
IOV-1050B *Bare board* **$19.95**

S-100 CPU

2810 Z-80* CPU - Cal Comp Sys

2/4 MHz Z-80A CPU with RS-232C serial I/O port and on-board MOSS 2.2 monitor PROM, front panel compatible.*

CPU-30400A *A & T* **$269.95**



THE BIG Z* - Jade

2 or 4 MHz switchable Z-80 CPU with serial I/O, accomodates 2708, 2716, or 2732 EPROM, baud rates from 75 to 9600*

CPU-30201K *Kit* **$145.00**
CPU-30201A *A & T* **$199.00**
CPU-30200B *Bare board* **$35.00**

CB-2 Z-80 CPU - S.S.M.

2 or 4 MHz Z-80 CPU board with provision for up to 8K of ROM or 4K of RAM on board, extended addressing, IEEE S-100, front panel compatible.

CPU-30300A *A & T* **$229.95**

SBC-200 - SD Systems

4 MHz Z-80 CPU with serial & parallel I/O ports, up to 8K of on-board PROM, software programmable baud rate generator, 1K of on-board RAM, Z-80 CTC.*

CPC-30200K *Kit* **$339.95**
CPC-30200A *Jade A & T* **$399.95**

S-100 Disk Controller

DOUBLE DENSITY - Cal Comp Sys

5¼" and 8" disk controller, single or double density, with on-board boot loader ROM, and free CP/M 2.2 and manual set.*

IOD-1300A *A & T* **$369.95**

DOUBLE-D - Jade

Double density controller with the inside track, on-board Z-80A, printer port, IEEE S-100, can function on an interrupt driven buss*

IOD-1200K *Kit* **$299.95**
IOD-1200A *8" A & T* **$389.95**
IOD-1205A *5¼" A & T* **$389.95**
IOD-1200B *Bare board* **$65.00**

VERSAFLOPPY II - SD Systems

New double density controller for both 8" & 5¼"

IOD-1160K *Kit* **$379.95**
IOD-1160A *Jade A & T* **$439.95**

Motherboards

ISO-BUS - Jade

Silent, simple, and on sale - a better motherboard

6 Slot (5¼" x 8½")

MBS-061B *Bare board* **$19.95**
MBS-061K *Kit* **$39.95**
MBS-061A *A & T* **$49.95**

12 Slot (9¾" x 8½")

MBS-121B *Bare board* **$29.95**
MBS-121K *Kit* **$69.95**
MBS-121A *A & T* **$89.95**

18 Slot (14½" x 8½")

MBS-181B *Bare board* **$49.95**
MBS-181K *Kit* **$99.95**
MBS-181A *A & T* **$139.95**

Card Cages

S-100 CARD CAGE - Jade

Metal cage with card guides & fan mounting

ENX-106001 *Six slot* **$29.95**

S-100 CARD CAGE - Vector

19" rack mountable, adjustable, holds 21 cards

VCT-CCK100 *Anodized Al* **$49.95**

SALE SALE SALE

S-100 I/O

S.P.I.C. - Jade

Our new I/O card with 2 SIO's, 4 CTC's, and 1 PIO

IOI-1045K	*2 CTC's, 1 SIO, 1 PIO*	$199.00
IOI-1045A	*A & T*	$259.00
IOI-1046K	*4 CTC's, 2 SIO's, 1 PIO*	$259.00
IOI-1046A	*A & T*	$319.00
IOI-1045B	*Bare board w/ manual*	$59.95
IOI-1045D	*Manual only*	$20.00

I/O-4 - S.S.M.

2 serial I/O ports plus 2 parallel I/O ports

IOI-1010K	*Kit*	$179.95
IOI-1010A	*A & T*	$249.95
IOI-1010B	*Bare board*	$35.00

TERMINATOR - S.S.M.

Active terminator for S-100 bus

TSX-195K	*Kit*	$29.95
TSX-195A	*A & T*	$54.95
TSX-195B	*Bare board*	$22.95

S-100 EXTENDER - Cal Comp Sys

Put those problem boards (the ones you probably bought from one of our competitors) within easy reach.

TSX-160A	*A & T*	$37.95

S-100 PROTO BOARD - Jade

Universal design, plated thru holes, gold fingers

TSX-140B	*Bare board*	$24.95

TERMINATOR & EXTENDER - C.C.S.

Can be used as both an S-100 extender and terminator

TSX-150K	*Kit*	$43.95

Diskettes

DISKETTES - Jade

Bargain prices on magnificent magnetic media

5¼" single sided, single density, box of 10		
MMD-5110103	*Soft sector*	$27.95
MMD-5111003	*10 sector*	$27.95
MMD-5111603	*16 sector*	$27.95
5¼" double sided, double density, box of 10		
MMD-5220103	*Soft sector*	$39.95
8" single sided, single density, box of 10		
MMD-8110103	*Soft sector*	$33.95
8" single sided, double density, box of 10		
MMD-8120103	*Soft sector*	$39.95
8" double sided, double density, box of 10		
MMD-8220103	*Soft sector*	$49.95

Video Monitors

9" B & W MONITOR - A.P.F.

High quality, high resolution video monitor

VDM-750900	*9" monitor*	$159.95

13" COLOR MONITOR - Zenith

The hi res color you've been promising yourself

VDC-201301		$449.00

12" GREEN SCREEN - NEC

20 MHz, P31 phosphor video monitor with audio, exceptionally high resolution - A fantastic monitor at a very reasonable price

VDM-651200	*12" monitor*	$259.95

Mainframes

MAINFRAME - Cal Comp Sys

12 slot S-100 mainframe with 20 amp power supply

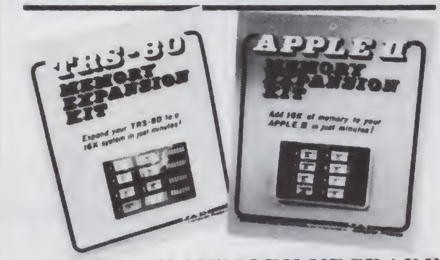
ENC-112105	*Kit*	$359.95
ENC-112106	*A & T*	$419.95

DISK MAINFRAME - NNC

Holds 2 8" drives and an 8 slot S-100 system. Attractive metal cabinet with 8 slot motherboard, power supply, fan, key switch, and other professional features

ENS-112320	*with 30 amp p.s.*	$699.95

Accessories-Apple/TRS-80



16K MEMORY UPGRADE

Add 16K of RAM to your TRS-80, Apple, or Exidy in just minutes. We've sold thousands of these 16K RAM upgrades which include the appropriate memory chips (as specified by the manufacturer), all necessary jumper blocks, fool-proof instructions, and our 1 year guarantee.

MEX-16100K	*TRS-80 kit*	$29.00
MEX-16101K	*Apple kit*	$29.00
MEX-16102K	*Exidy kit*	$29.00

DISK DRIVE for APPLE

5¼" disk drive with controller for your Apple

MSM-12310C	*with controller*	$499.95
MSM-123101	*w/out controller*	$375.00

DISK DRIVES for TRS-80

23% more storage, 8 times faster, 40 track with free patch, 120 day warranty, includes case, power supply, and cable

MSM-12410C	*Save $125.00 !!!*	$299.95

DOS 3.3 UPGRADE - Apple

Upgrade your old DOS to the improved 3.3

IOD-2233A	*Complete kit*	$64.95

APPLE STICK - Micromate

Joy stick with pots for Apple II

SYA-1510A	*A & T*	$35.95

Z-80* CARD for APPLE

Z-80 CPU card with CP/M 2.2 for your Apple*

CPX-30800A	*A & T*	$279.95

AIO - S.S.M.

Parallel & serial interface for your Apple

IOI-2050K	*Kit*	$155.95
IOI-2050A	*A & T*	$194.95

PRINTER INTERFACE - C.C.S.

Centronics type I/O card w/ firmware

IOI-2041A	*A & T*	$99.95

APPLE CLOCK - Cal Comp Sys

Real time clock w/ battery back-up

IOK-2100A	*A & T*	$109.95

Modems

★★★★★★★★★★★★★★★★★★★★★★★★★

LEX-11 MODEM - Lexicon

A real star! 300 baud, answer/originate, RS 232C

IOM-5511A	*Best buy !!!*	$128.00

★★★★★★★★★★★★★★★★★★★★★★★★★

NOVATION CAT

300 baud, answer/originate acoustic modem

IOM-5200A	*1 year warranty*	$179.00

D-CAT 300 baud, direct connect modem

IOM-5201A	*Special sale price*	$189.00

AUTO-CAT Auto answer/originate, direct connect

IOM-5230A	*Special sale price*	$239.95

MICROMODEM - D.C. Hayes

Auto answer/dial modem card for Apple or S-100

IOM-2010A	*Apple modem*	$349.95
IOM-1100A	*S-100 modem*	$375.00

MICRONET MODEM - Micromate

Direct connect with extra features - a best buy

IOM-2020A	*Best Apple modem*	$275.00

* Z-80, Z-80A, and the letter Z are recognized trademarks of Zilog, Inc. *CP/M is a registered trademark of Digital Research Corp. *CBASIC is a trademark of Compiler Systems, Inc.

Single Board Computers

AIM-65 - Rockwell

6502 computer with alphanumeric display, printer, & keyboard, and complete instructional manuals

CPK-50165	*1K AIM*	$374.95
CPK-50465	*4K AIM*	$449.95
SFK-74600008E	*8K BASIC ROM*	$99.95
SFK-64600004E	*4K assembler ROM*	$84.95
PSX-030A	*Power supply*	$64.95
ENX-000002	*Enclosure*	$49.95

4K AIM, 8K BASIC, power supply, & enclosure

Special package price **$625.00**

Z-80* STARTER KIT - SD Systems

Complete Z-80 computer with RAM, ROM, I/O, display, keyboard, manual, and kluge area.*

CPS-30010K	*Kit*	$369.95
CPS-30010A	*Jade A & T*	$459.95

MICROPROCESSORS

Z-80	10.95
Z-80A	12.95
6502	11.50
6800	11.95
6802	17.95
6809	39.95
8035	24.00
8080A	6.59
8085	15.95
8748	59.95

Z-80 SUPPORT

3881 *PIO*	9.50
3881-4 *PIO-4 MHz*	14.50
3882 *CTC*	9.50
3882-4 *CTC-4 MHz*	14.95
3883 *SIO*	29.50
3884 *SIO*	49.50

BAUD RATE GENERATORS

MC14411	10.00
1.843 MHz xtal	4.95

UARTS

AY5-1013A	5.25
AY3-1014A	8.25
TR1602B	5.25
TMS6011	5.95
IM6402	9.00

6800 SUPPORT

6821P	5.95
6828P	11.95
6834P	22.50
6840P	18.75
6850P	4.80
6852P	5.79
6875L	7.40
68488P	25.00

PROMS

2708 *450ns*	6.25
10 for $4.90 ea	
2716 *12.5v*	11.95
2716 *5v*	11.95
10 for $8.90 ea	
2532 *5v*	39.95
2732 *5v*	39.95
2758 *5v*	9.95

RAMS

21L02 *2 MHz*	1.25
21L02A *4 MHz*	1.50
2114L *2 MHz*	3.75
2114LA *4 MHz*	3.95
4116	4.25
4164 *64K x1*	59.95
5257 *2 MHz*	6.75
5257A *4 MHz*	7.25
MK4118	18.95

SUPPORT DEVICES

8212	3.25
8214	4.65
8216	2.95
8224	3.25
8224-4	5.75
8226	3.85
8228	4.95
8238	4.95
8243	8.00
8250	14.95
8251	6.50
8253	17.95
8255	6.50
8257	19.95
8259	17.95
8275	49.95
8279	15.95

PLACE ORDERS TOLL FREE

Continental U.S.	*Inside California*
800-421-5500	**800-262-1710**

For Technical Inquiries or Customer Service call

213-973-7707

48

JADE Computer Products

4901 W. Rosecrans, Hawthorne, Ca 90250

TERMS OF SALE: Cash, checks, credit cards, or Purchase Orders from qualified firms and institutions. Minimum order $15.00. California residents add 6% tax. Minimum shipping and handling charge $3.00. **Pricing and availability subject to change without notice.**

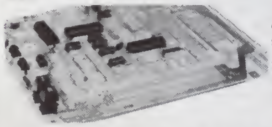
*Start learning and computing for only **$129.95** with a Netronics 8085-based computer kit. Then expand it in low-cost steps to a business/development system with 64k or more RAM, 8" floppy disk drives, hard disks and multi-terminal I/O.*

THE NEW EXPLORER/85 SYSTEM

Special! *Full 8" floppy, 64k system for less than the price of a mini! Only* $1499.95!

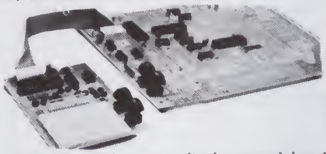
(Also available wired & tested, $1799.95)

Imagine — for only $129.95 you can own the starting level of Explorer/85, a computer that's expandable into full business/development capabilities — a computer that can be your beginner system, an OEM controller, or an IBM-formatted 8" disk small business system. From the first day you own Explorer/85, you begin computing on a significant level, and applying principles discussed in leading computer magazines. Explorer/85 features the advanced Intel 8085 cpu, which is 100% compatible with the older 8080A. It offers onboard S-100 bus expansion, Microsoft BASIC in ROM, plus instant conversion to mass storage disk memory with standard IBM-formatted 8" disks. All for only $129.95, plus the cost of power supply, keyboard/terminal and RF modulator if you don't have them (see our remarkable prices below for these and other accessories). With a Hex Keypad/display front panel, Level "A" can be programmed with no need for a terminal, ideal for a controller, OEM, or a real low-cost start.



Level "A" is a complete operating system, perfect for beginners, hobbyists, industrial controller use. $129.95

Full 8" disk system for less than the price of a mini (shown with Netronics Explorer/85 computer and new terminal). System features floppy drive from Control Data Corp., world's largest maker of memory storage systems (not a hobby brand!)



Level "A" With Hex Keypad/Display.

LEVEL "A" SPECIFICATIONS

Explorer/85's Level "A" system features the advanced Intel 8085 cpu, an 8355 ROM with 2k deluxe monitor/operating system, and an advanced 8155 RAM I/O . . . all on a single motherboard with room for RAM/ROM/PROM/EPROM and S-100 expansion, plus generous prototyping space.

PC Board: Glass epoxy, plated through holes with solder mask. • **I/O:** Provisions for 25-pin (DB25) connector for terminal serial I/O, which can also support a paper tape reader . . . cassette tape recorder input and output . . . cassette tape control output . . . LED output indicator on SOD (serial output) line . . . printer interface (less drivers) . . . total of four 8-bit plus one 6-bit I/O ports. • **Crystal Frequency:** 6.144 MHz. • **Control Switches:** Reset and user (RST 7.5) interrupt . . . additional provisions for RST 5.5, 6.5 and TRAP interrupts onboard. • **Counter/Timer:** Programmable, 14-bit binary. • **System RAM:** 256 bytes located at F800, ideal for smaller systems and for use as an isolated stack area in expanded systems . . . RAM expandable to 64K via S-100 bus or 4k on motherboard.

System Monitor (Terminal Version): 2k bytes of deluxe system monitor ROM located at FØØØ, leaving ØØØØ free for user RAM/ROM. Features include tape load with labeling . . . examine/change contents of memory . . . insert data . . . warm start . . . examine and change all registers . . . single step with register display at each break point, a debugging/training feature . . . go to execution address . . . move blocks of memory from one location to another . . . fill blocks of memory with a constant . . . display blocks of memory . . . automatic baud rate selection to 9600 baud . . . variable display line length control (1-255 characters/line) . . . channelized I/O monitor routine with 8-bit parallel output for high-speed printer . . . serial console in and console out channel so that monitor can communicate with I/O ports.

System Monitor (Hex Keypad/Display Version): Tape load with labeling . . . tape dump with labeling . . . examine/change contents of memory . . . insert data . . . warm start . . . examine and change all registers . . . single step with register display at each break point . . . go to execution address. Level "A" in this version makes a perfect controller for industrial applications, and is programmed using the Netronics Hex Keypad/Display. It is low cost, perfect for beginners.

HEX KEYPAD/DISPLAY SPECIFICATIONS

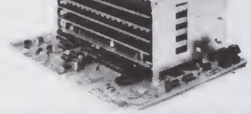
Calculator type keypad with 24 system-defined and 16 user-defined keys. Six digit calculator-type display, that displays full address plus data as well as register and status information.

LEVEL "B" SPECIFICATIONS

Level "B" provides the S-100 signals plus buffers/drivers to support up to six S-100 bus boards, and includes: address decoding for onboard 4k RAM expansion selectable in 4k blocks . . . address decoding for onboard 8k EPROM expansion selectable in 8k blocks . . . address and data bus drivers for onboard expansion . . . wait state generator (jumper selectable), to allow the use of slower memories . . . two separate 5 volt regulators.

LEVEL "C" SPECIFICATIONS

Level "C" expands Explorer/85's motherboard with a card cage, allowing you to plug up to six S-100 cards directly into the motherboard. Both cage and card are neatly contained inside Explorer's deluxe steel cabinet. Level "C" includes a sheet metal superstructure, a 5-card, gold plated S-100 extension PC board that plugs into the motherboard. Just add required number of S-100 connectors.



Explorer/85 With Level "C" Card Cage.

LEVEL "D" SPECIFICATIONS

Level "D" provides 4k of RAM, power supply regulation, filtering decoupling components and sockets to expand your Explorer/85 memory to 4k (plus the original 256 bytes located in the 8155A). The static RAM can be located anywhere from ØØØØ to EFFF in 4k blocks.

LEVEL "E" SPECIFICATIONS

Level "E" adds sockets for 8k of EPROM to use the popular Intel 2716 or the TI 2516. It includes all sockets, power supply regulator, heat sink, filtering and decoupling components. Sockets may also be used for 2k x 8 RAM IC's (allowing for up to 12k of onboard RAM).

DISK DRIVE SPECIFICATIONS

- 8" CONTROL DATA CORP. professional drive.
- LSI controller.
- Write protect.
- Single or double density.
- Data capacity: 401,016 bytes (SD), 802,032 bytes (DD), unformatted.
- Access time: 25ms (one track).

DISK CONTROLLER/ I/O BOARD SPECIFICATIONS

- Controls up to four 8" drives.
- 1771A LSI (SD) floppy disk controller.
- Onboard data separator (IBM compatible).
- 2 Serial I/O ports
- Autoboot to disk system when system reset.
- 2716 PROM socket included for use in custom applications.
- Onboard crystal controlled.
- Onboard I/O baud rate generators to 9600 baud.
- Double-sided PC board (glass epoxy.)

DISK DRIVE CABINET/POWER SUPPLY

- Deluxe steel cabinet with individual power supply for maximum reliability and stability.

ORDER A COORDINATED EXPLORER/85 APPLICATIONS PAK!

Beginner's Pak (Save $26.00!) — Buy Level "A" (Terminal Version) with Monitor Source Listing and AP-1 5-amp Power Supply: (regular price $199.95), now at SPECIAL PRICE: **$169.95** plus post. & insur.

Experimenter's Pak II (Save $53.40!) — Buy Level "A" (Hex Keypad/Display Version) with Hex Keypad/Display, Intel 8085 User Manual, Level "A" Hex Monitor Source Listing, and AP-1 5-amp Power Supply: (regular price $279.35), all at SPECIAL PRICE: **$219.95** plus post. & insur.

Special Microsoft BASIC Pak (Save $103.00!) — Includes Level "A" (Terminal Version), Level "B", Level "D" (4k RAM), Level "E", 8k Microsoft in ROM, Intel 8085 User Manual, Level "A" Monitor Source Listing, and AP-1 5-amp Power Supply: (regular price $439.70), now yours at SPECIAL PRICE: **$329.95** plus post. & insur.

ADD A TERMINAL WITH CABINET, GET A FREE RF MODULATOR: Save over $114 at this SPECIAL PRICE: **$499.95** plus post. & insur.

Special 8" Disk Edition Explorer/85 (Save over $104!) — Includes disk-version Level "A", Level "B", two S-100 connectors and brackets, disk controller, 64k RAM, AP-1 5-amp power supply, Explorer/85 deluxe steel cabinet, cabinet fan, 8" SD/DD disk drive from famous CONTROL DATA CORP. (not a hobby brand!), drive cabinet with power supply, and drive cable set-up for two drives. This package includes everything but terminal and printers (see coupon for them). Regular price $1630.30, all yours in kit at SPECIAL PRICE: **$1499.95** plus post. & insur. Wired and tested, only **$1799.95.**

Special! Complete Business Software Pak (Save $625.00!) — Includes CP/M 2.0, Microsoft BASIC, General Ledger, Accounts Receivable, Accounts Payable, Payroll Package: (regular price $1325), yours now at SPECIAL PRICE: **$699.95.**

Please send the items checked below:

- ☐ **Explorer/85 Level "A" kit** (Terminal Version) . . . **$129.95** plus $3 post. & insur.
- ☐ **Explorer/85 Level "A" kit** (Hex Keypad/Display Version) . . **$129.95** plus $3 post. & insur.
- ☐ **8k Microsoft BASIC** on cassette tape. **$64.95** postpaid.
- ☐ **8k Microsoft BASIC in ROM kit** (requires Levels "B", "D" and "E") . . . **$99.95** plus $2 post. & insur.
- ☐ **Level "B" (S-100) kit** . . . **$49.95** plus $2 post. & insur.
- ☐ **Level "C" (S-100 6-card expander) kit** . . . **$39.95** plus $2 post. & insur.
- ☐ **Level "D" (4k RAM) kit** . . . **$69.95** plus $2 post. & insur.
- ☐ **Level "E" (EPROM/ROM) kit** . . . **$5.95** plus 50¢ p&h.
- ☐ **Deluxe Steel Cabinet** for Explorer/85 . . . **$49.95** plus $3 post. & insur.
  - ☐ Fan For Cabinet . . . **$15.00** plus $1.50 post. & insur.
- ☐ **ASCII Keyboard/Computer Terminal kit:** features a full 128 character set, u&l case; full cursor control; 75 ohm video output; convertible to baudot output; selectable baud rate, RS232-C or 20 ma. I/O, 32 or 64 character by 16 line formats, and can be used with either a CRT monitor or a TV set (if you have an RF modulator) . . . **$149.95** plus $3.00 post. & insur.
  - ☐ **DeLuxe Steel Cabinet** for ASCII keyboard/terminal . . . **$19.95** plus $2.50 post. & insur.
- ☐ **New! Terminal/Monitor:** (See photo) Same features as above, except 12" monitor with keyboard and terminal is in deluxe single cabinet; kit . . . **$399.95** plus $7 post. & insur.
- ☐ **Hazeltine terminals:** Our prices too low to quote — **CALL US**
- ☐ **Lear-Sigler terminals/printers:** Our prices too low to quote: **CALL US**
- ☐ **Hex Keypad/Display kit** . . . **$69.95** plus $2 post. & insur.
- ☐ **AP-1 Power Supply Kit** ±8V @ 5 amps) in deluxe steel cabinet . . . **$39.95** plus $2 post. & insur.
- ☐ **Gold Plated S-100 Bus Connectors** . . . **$4.85** each, postpaid.
- ☐ **RF Modulator kit** (allows you to use your TV set as a monitor) . . . $8.95 **postpaid.**
- ☐ **16k RAM kit** (S-100 board expands to 64k) . . . **$199.95** plus $2 post. & insur.
- ☐ **32k RAM kit** . . . **$299.95** plus $2 post. & insur.
- ☐ **48k RAM kit** . . . **$399.95** plus $2 post. & insur.
- ☐ **64k RAM kit** . . . **$499.95** plus $2 post. & insur.
- ☐ **16k RAM Expansion kit** (to expand any of the above in 16k blocks up to 64k) . . . **$99.95** plus $2 post. & insur. each.
- ☐ **Intel 8085 cpu Users' Manual** . . . **$7.50** postpaid.
- ☐ **12" Video Monitor** (10MHz bandwidth) . . . **$139.95** plus $5 post. & insur.
- ☐ **Beginner's Pak** (see above) **$169.95** plus $4 post. & insur.
- ☐ **Experimenter's Pak** (see above) . . . **$219.95** plus $6 post. & insur.
- ☐ **Special Microsoft BASIC Pak Without Terminal** (see above) . . . **$329.95** plus $7 post. & insur.
- ☐ Same as above, plus **ASCII Keyboard Terminal With Cabinet, Get Free RF Modulator** (see above) . . . **$499.95** plus $10 post. & insur.
- ☐ **Special 8" Disk Edition Explorer/85** (see above) . . . **$1499.95** plus $26 post. & insur.
  - ☐ **Wired & Tested** . . . **$1799.95** plus $26 post. & insur.
- ☐ **Extra 8" CDC Floppy Drives** . . . **$499.95** plus $12 post. & insur.
  - ☐ **Cabinet & Power Supply For Drive** . . . **$69.95** plus $3 post. & insur.
- ☐ **Drive Cable Set-up For Two Drives** . . . **$25** plus $1.50 post. & insur.
- ☐ **Disk Controller Board With I/O Ports** . . . **$199.95** plus $2 post. & insur.
- ☐ **Special: Complete Business Software Pak** (see above) . **$699.96** postpaid.
  - SOLD SEPARATELY:
  - ☐ **CP/M 1.4** . . . **$100** postpaid.
  - ☐ **CP/M 2.0** . . . **$150** postpaid.
  - ☐ **Microsoft BASIC** . . . **$325** postpaid.
- ☐ **Intel 8085 cpu User Manual** . . . $7.50 **postpaid.**
- ☐ **Level "A" Monitor Source Listing** . . . **$25** postpaid.

Dept. KB3

Continental U.S.A. Credit Card Buyers Outside Connecticut

CALL TOLL FREE: 800-243-7428

To Order From Connecticut Or For Technical Assistance, call (203) 354-9375

Total Enclosed (Conn res. add sales tax) $ ________

Paid By:

☐ Personal Check ☐ Cashier's Check/Money Order

☐ VISA ☐ Master Charge (Bank No. __ __ __ __)

Acct. No. ____________ Exp. Date ______

Signature ____________

Print Name ____________

Address ____________

City ____________

State ________ Zip ________

NETRONICS Research & Development Ltd.
333 Litchfield Road, New Milford, CT 06776

✓ Reader Service—see page 178

Best Prices and Delivery!

SUPERBRAIN by Intertec

Self contained computer with dual disks and two RS232C ports. Complete with CP/M 2.2 and BASIC.

32K Double Density, List $2995 . **$2685**
64K Double Density, List $3345 .... $2883
64K Upgraded to Quad Density w/Special MiniMicroMart Warranty .... ***SPECIAL $3395***

VIDEO TERMINALS

NEW **EMULATOR (Intertec)**, List $895 $ 749
NEW INTERTUBE III List $895 ... **ONLY $ 749**
SOROC 120, List $995 ...... ***SPECIAL $ 729***
IQ140, List $1495 .... ***SPECIAL $1149***
PERKIN-ELMER 550, List $997 ...... $ 799
with anti-glare screen, $1027 .... $ 829
HAZELTINE 1410, List $900 .... $ 749
1420 .... $ 849
1500, List $1225 .... $ 879
1510, List $1395 .... $1089
1520, List $1650 .... $1389
LEAR SIEGLER ADM3A, Assembled .. $ 849
TELEVIDEO 912C, List $950 .... $ 789
920C List $1030 .... $ 849
950C, List $1195 .... **NEW $1039**

PRINTERS

ANADEX DP-8000 .... $ 849
DP-9500, List $1650 .... $1399
DP-9501, List $1650 .... $1399
PAPER TIGER IDS-445, List $995 ...... $ 695
w/graphics op., incl. buffer, $1195 .. $ 789
PAPER TIGER IDS-460, List 1149
PAPER TIGER IDS-460G List 1199
NEW IDS PAPERTIGER 460 . List 1295 $1149
NEW IDS PAPERTIGER 460G List $1394 $1199
NEC Spinwriters .... Call for Price
TELETYPE 43 KSR $1087
CENTRONICS
730-1 parallel interface .. **NEW LOW $ 649**
737 parallel interface . **SUPER VALUE $ 829**
779 w/Tractor, List $1350 .... $1049
703 w/Tractor, VFU, List $2975 ...... $1695
704 w/Tractor, VFU, List $2350 ...... $1595

TI 810 Basic, List $1895 .... $1695
810 serial & Centronics-style parallel interface, List $1940 ...... $1735
810 w/full ASCII (U/LC), Vertical Forms Control, Compressed Print ... $1895
TI 820 KSR, List $2165 .... $1895
TI 745 w/full ASCII, List $1695 .... $1399
COMPRINT 912 w/parallel interface .. $ 559
912 w/serial interface, List $699 ...... $ 589
AXIOM IMP I .... $ 699
MICROTEK, List $750 .... $ 675
OKIDATA Microline 80, List $949 ..... $ 599
Tractor Feed Option $109
RS232 Serial Interface .... $ 99

NORTH STAR HORIZON®

HORIZON 1 ASSEMBLED & TESTED

32K, Double Density, List $2695 .... $2279
32K, Quad Density, List $2995 .... $2539

HORIZON 2 ASSEMBLED & TESTED

32K, Double Density, List $3095 ...... $2619
32K, Quad Density, List $3595 .... $3049
48K, Double Density, List $3590 . .... $3039
48K, Quad Density, List $4090 .... $3469
64K, Double Density, List $3830 ...... $3239
64K, Quad Density, List $4330 .... $3669

FLOPPY DISK SYSTEMS

NORTH STAR MDS-A
Assembled, List $899 ...... ***SPECIAL $ 719***

MORROW THINKER TOYS® Discus 2D, List $1199 .... **OUR PRICE $ 998***
Discus 2D, dual-drive, List $1994 ..... $1694*
Discus 2 + 2, A&T, List $1549 .... $1319*
Dual Discus 2 + 2, A&T, List $2748 ... $2335*
*Now includes CP/M® 2.2

Prom Programers

SSM PB1 Kit List $125 .... $106
SSM PB1 A&T List $190............. $161
SD Computer Prom 100 Kit $236....... $201
SD Computer Prom 100A&T List $311 .. $264

FLOPPY DISK CONTROLLER BOARDS

NORTH STAR, DD, Assembled, List $499 .... $399
MORROW Disk Jockey 1, A&T ($213) . $189
Disk Jockey 2D, A&T, List $479 .... $429
SD Versafloppy 1, Kit, List $284 ...... $239
Versafloppy II, DD Kit, List $413 ..... $349
Versafloppy II, DD, A&T, List $507 ... $429
DELTA double density A&T ($385) $345
CONDUCTOR, double density A&T ... $269
INTERSYSTEMS FDC-2, A&T, $495 ... $439
MICROMATION Doubler, DD, A&T ... $429
TARBELL Floppy Disk Interface Kit $199
double density, A&T, List $495 .... $444

ESCON CONVERSION FOR IBM SELECTRIC

Complete w/microprocessor controller and power supply. Factory built. User installs solenoid assembly or it can be done at Escon factory at nominal cost.

Parallel (TRS-80, Sorcerer, etc.), $575 $514
RS232 Standard Serial, List $599 534
IEEE-488 (for PET), List $660 584
TRS-80 Cable 25

CALIFORNIA COMPUTER SYSTEMS

Z80 CPU BOARDS List $299 .... $269
DISK CONTROLLER 2422 List $399 .... $359
16K Static, A&T, List $349.95 .... $259
32K STATIC List $710 .... $599
64K DYNAMIC BOARD List $699 ...... $589

CPU BOARDS

(assembled unless noted)

NORTH STAR Z80A (ZPB-A/A), $299 $254
CROMEMCO 4 MHz (ZPU-W), List $395 $335
4 MHz (SCC-W), List $450 .... $382
INTERSYSTEMS (formerly Ithaca Audio)
new Series II Z 80, 4 MHz, List $395 ... $349
SSM CB1 8080 A&T List $252 .... $214
CB1A Kit, List $183 .... $156
CB2 Z-80, A&T, List $344 .... $289
CB2 Kit, List $60 .... $221
DELTA Z-80, with I/O $289
SD SBC-100, List 413 .... $349
SBC-100 Kit, List $341 .... $289
SBC-200, List $471 .... $399
SBC-200 Kit, List $373 .... $317

MEMORY BOARDS

32K SD ExpandoRAM Kit
CALL FOR PRICES

NORTH STAR 16K Dynamic RAM Board, A&T (RAM-16-A/A), List $499 .... $420
32K A&T (RAM-32/A), List $739 ...... $620
CROMEMCO 16KZ-W, List $495 ...... $419
64KZ-W, List $1795 .... $1269
MEASUREMENT SYSTEMS & CONTROLS
(Guaranteed performance, incl. labor/parts 1 yr)
DM6400 64K Board w/all 64K, $795 ... $659
DM4800 with 48K, List $695 .... $650
DM3200 with 32K, List $595 .... $509
DMB6400 64K Board w/all 64K $859
DMB4800 with 48K $789
MORROW SuperRAM — all static, all A&T
16K, 4 MHz or 2 MHz, List $349 .... $299
32K, 4 MHz, List $699 .... $629
16K Memory Master, List $399 .... $339
24K Memory Master, List $549 .... $465
INTERSYSTEMS (formerly Ithaca Audio)
8K Static 2 MHz, A&T, List $165 ...... $149
8K Static 4 MHz, A&T, List $195 ...... $176
16K Static 2 MHz, A&T, List $475 ..... $427
16K Static 4 MHz, A&T, List $495 ..... $445
64K Dynamic, List $995 .... $895
CALIFORNIA COMPUTER
16K Static, A&T, List $349.95 .... $259

VIDEO BOARDS

I/O Mapped
SD COMPUTER VDB-8024, kit, List $437 .369
Assembled, List $556 .... $469
XITEX SCT-100K, Kit .... ONLY $169.95
SCT-100A Assembled $189.95
SSM VB2 I/O, Kit, List $199 .. ..$169
Assembled & Tested, List $269 ....$229
Memory Mapped
SSM VB1C, 16x64, Kit, List $179 ....$152
Assembled & Tested, List $242 ....$206
SSM VB3, 80-Char., 4MHz, Kit, List$425 . . $359
4 MHz, A&T, List $499 ....$424
INTERSYSTEMS, 16x64, A&T, List $165 $149
*Subject to change.

NEW CROMEMCO 16FDC DOUBLE DENSITY DISK CONTROLLER

List $595 ...... **OUR PRICE $505**

SHIPPING AND INSURANCE. Add $2.50 for boards, $6 for Selectric Converter or Floppy Disk Drives, $7.50 for Floppy Disk Systems, $15 for Horizon. SHIPPED FREIGHT COLLECT: SuperBrain, Centronics and T.I. printers. Contact us for shipping information on other terminals and printers.

Above prices reflect a 2% cash discount (order prepaid prior to shipment). Add 2% to prices for credit card orders, C.O.D.'s, etc. Prices are subject to change and offers subject to withdrawal without notice.

— WRITE FOR FREE CATALOG —

MiniMicroMart, Inc.

227

1618 James Street, Syracuse NY 13203 (315) 422-4467 TWX 710-541-0431

Terminals and Printers!

TELEVIDEO TVI-912C

Upper and lower case, 15 baud rates: 75 to 19,000 baud, dual intensity, 24 x 80 character display, 12 x 10 resolution. Numeric pad. Programmable reversible video, auxiliary port, self-test mode, protect mode, block mode, tabbing, addressable cursor. Microprocessor controlled, programmable underline, line and character insert/delete. "C" version features typewriter-style keyboard. List $950

OUR PRICE $789

920C (with 11 function keys, 6 edit keys and 2 transmission mode keys, List $1030

ONLY $849

New 950C $1195 .... **$1039**

Intertec EMULATOR

Software compatible with a Soroc IQ-120, Hazeltine 1500, ADM-3A or DEC VT-52. Features block mode transmission and printer port; 12" anti-glare screen; 18-key numeric keypad; full cursor control. List $895

OUR PRICE $749

NEW INTERTUBE III

List $995 **ONLY $749**

12" display, 24 x 80 format, 18-key numeric keypad, 128 upper/lower case ASCII characters. Reverse video, blinking, complete cursor addressing and control. Special user-defined control function keys, protected and unprotected fields. Line insert/delete and character insert/delete editing, eleven special line drawing symbols.

SOROC

IQ-120

List $995

SPECIAL $729

IQ-140 List $1495

SPECIAL $1149

HAZELTINE

1500 ONLY $879

1410 w/numeric keypad, List $900	**$749**
1420 w/lower case and numeric pad	**849**
1510, List $1395	**1089**
1520, List $1650	**1389**

NEC SPINWRITER™

Terminal/Keyboard as well as RO Printer Only models available.

CALL FOR PRICES!

CENTRONICS PRINTERS

NEW 730-1, parallel, friction, tractor ..	**$649**
NEW 730-3 RS-232 friction, tractor	**$699**
NEW 737-1 parallel, friction, tractor	**$829**
NEW 737-3 RS-232 friction, tractor	**$899**
779-2 w/tractor (Same as TRS-80 Line Printer I), List $1350	**$849**
704-11 parallel 180 cps	**$1698**
704-9 RS232 serial version, $2350 ..	**$1595**

TI-810

TI-810 Basic Unit, $1895 .	**ONLY $1595**
TI-810 w/full ASCII (Lower case), vertical forms control, and compressed print .	**$1795**
TI-745 Complete printing terminal with acoustic coupler, List $1695	**$1399**

ANADEX DP9500/DP9501 PRINTERS

DP-9500, List $1650	**$1349**
DP-9501, List $1650	**$1349**

OKIDATA Microline 80 **ONLY $599**

Tractor Feed Option	**$109**
Serial interface	**$ 99**
AXIOM IMP I	**$699**
COMPRINT 912 w/parallel interf.	**$559**
912 w/serial interface, List $699	**$589**
MICROTEK, List $750	**$675**
NEW MODELS!!!	
DP 9500	**$1349**
DP 9501	**$1349**

PAPER TIGER®

IDS-445 Paper Tiger, ...	**$695**
w/graphics option, incl. buffer, $1194 ..	**$789**
TRS-80 cable	**$ 45**
NEW IDS PAPERTIGER 460 List $1295	***$1099***
NEW IDS PAPERTIGER 460G List $1394	**$1149**

NEW IDS 460
QUALITY PRINTING AT MATRIX SPEED—LOGIC SEEKING PROPORTIONAL SPACING
w/auto text justification

Above prices reflect a 2% cash discount (order prepaid prior to shipment). Add 2% to prices for credit card orders, C.O.D.'s, etc. Prices are f.o.b. shipping point. Prices are subject to change and offers subject to withdrawal without notice. **WRITE FOR FREE CATALOG.**

✓50

MiniMicroMart, Inc.

VISA

master charge

1618 James Street, Syracuse NY 13203 (315) 422-4467 TWX 710-541-0431

MICROCOMPUTING T.M. LIST OF ADVERTISERS

Reader Service Number	Advertiser	Page
121	AB Computers	75
273	ABM Products	110
39	AM Electronics	141
232	Aardvark Technical Services	70
91	Aardvark Technical Services	147
56	American Square Computers	185
314	Apple-jack	183
334	Applied Business Computers	41,64
192	Audio Video Systems	193
193	Aurora Software	40
96	Automated Equipment, Inc.	98
55	Automated Simulations	17
495	Axlon, Inc.	180
99	John Bell Engineering	153
489	John Bell Engineering	183
110	CFR Associates	181
256	CPU Shop	97
79	C&S Electronics Mart, Ltd.	99
356	J.A. Cambron Company Inc.	40
26	Cherry Engineering	70
97	Commerce Tours International Inc.	64, 122
90	CompuCover	60
43	Compumart	166,167
486	CompuSoCo	186
•	Computer Broker	63
320	Computer Case Company	25
278	Computer Connections	111
475	Computer Control Systems Inc.	186
133	Computer CTY	127
18	Computer Design Labs	43
120	Computer Discount of America	62
152	Computer Distributors	140
80	Computer Services	119
49	Computer Services	188
36	Computer Shopper	183
283	The Computer Stop	58
105	The Computer Stop	190
493	Computer Textile Inc.	180
6	Computronics	105
297	Concord Computer Components	174
271	Coosol, Inc.	115
492	Correlation Systems	180
228	Cuddly Software	193
141	Custom Electronics, Inc.	134
474	Cyber Associates Inc.	188
•	Cybernetics	114
293	D & N Micro Products	70
24	The Datak Corporation	189
471	Datasoft	188
63	Davis Systems, Inc.	119
491	Digital Laboratories Inc.	184
•	Digital Research Computers	170,171
•	Digital Research Parts	115
61	Digital Systems Engineering	96
250	Discount Software Group	66
•	Disk Supply Company	111
155	Dorsett Educational Systems Inc.	114
34	Dr. Daley	109
88	Dr. Daley	148
87	Dwo Quong Fok Lok Sow	150
494	ETC Corporation	182
482	E-Z Software	188
178	Eastern House Software	182
82	Ecosoft	184
•	Electravalue Industrial	104
93	Electronic Specialists	75
272	Ellis Computing	89
254	Erickson Communications	127
57	Exatron	134
70	FMG Corporation	55
131	FireFlyer Computing	64
191	Floppy Disk Services	49
351	General Peripherals	66
22	Gimix Inc.	10
42	Godbout Electronics	163
236	Heath Company	15
487	Hewlett-Packard	180
340	High Sierra Software, Inc.	136
168	Honders Inc.	17
33	Human Engineered Software	49
209	Ian Electronics	10
40	Instant Software	26,27,51,61,128,129,150,182
77	Integrand Research Corp.	79
76	Intelligent Systems Corporation	71
151	Interface, Inc.	85
183	Interface Technology	57
235	Interlude	82
127	International Business Forms Inc.	93
195	Interpretive Education	39
3	Intertec Data	3
343	Intertec Data	67
180	J.E.S. Graphics	184
92	J.P.C. Products	188
126	JR Inventory Company	151
48	Jade Computer Products	172,173
41	Jameco Electronics	164,165
164	Jim Pak	191
247	Joe Computer	85
222	Kalglo Electronics Company Inc.	54
223	Key Bits Inc.	85
•	Kilobaud Microcomputing	20,31,81,107,158-161,184,194
198	LNW Research	127
355	Leading Edge Products	9
483	Richard Leary	189
23	Level IV Products, Inc.	42
234	Magnolia Microsystems	41
165	Charles Mann & Associates	109
84	Mark Gordon Computers	108
333	Meta Media Productions Inc.	7
161	Meta Technologies Corporation	19
108	Micro Architect, Inc.	101
476	Micro Architect, Inc.	189
167	Micro Computer Industries	37
260	The Microcomputer Warehouse	66
68	Micromail	41
100	Micro Management Systems	187
479	Micro Mike's Inc.	189
347	Micromint	104
335	Micro Products Unlimited	64
338	Micro Products Unlimited	92
488	Microsette	182
•	Midwest Scientific Instruments	156,157
	Mikos	169
255	Miller Microcomputer Services	24
227	Mini Micro Mart	176
50	Mini Micro Mart	177
78	Minis & Micros, Inc.	70
472	Mosaic Electronic	188
•	Mullen Computer Products	89
81	Multi Business Computer Systems	179
•	NRI Schools	47
•	National Computer Conference	137
•	National Computer Shows	142,143
•	Netronics R & D Ltd.	36,134,175
21	OK Machine & Tool	75
62	OK Machine & Tool	79
4	Ohio Scientific	C IV
130	Olensky Brothers, Inc.	119
89	Omega Sales Company	123
140	Omnitek Systems	54
484	Open Systems	186
29	Optimal Technology, Inc.	103
310	Orange Micro	179
149	Osborne/McGraw Hill	21
106	PAIA	25
19	Paccom	101
274	Pacific Exchanges	70
246	Pacific Exchanges	109
153	Pacific Office Systems	111
71	Pan American Electronics	42
98	Peek-65	92
13,14,16,69,204,245	Percom Data Company, Inc.	C II
478	Percom Data Company Inc.	186
266	Perry Oil & Gas	101
11	Programma International	35
277	Programma International	135
202	Progressive Computing	114
44	Quest Electronics	162
188	RAC Products	89
46	RKS Enterprises, Inc.	25
52	RNB Enterprises Inc.	168
101	Racet Computes	181
•	Radio Shack	C III
142	Random Access, Inc.	79
•	Realty Software Company	70
20	The Robb Report	146
74	Rondure Company	40
496	SSM Microcomputer Products	183
477	Serendipity Systems Inc.	189
208	Service Technologies Inc.	93
12	Simutek	132
67	Sirius Systems	149
132	68 Micro Journal	70
66	Skyles Electric Works	154,155
257	Snapp, Inc.	59
473	Software Resources Inc.	189
39	The Software Toolworks	33
306	Spectrum Software	133
288	The Stocking Source	13
237	Synergetic Solutions	179
481	Syntax Corporation	186
498	Syntest Corporation	182
189	Tab Sales Company	53
350	Tang Data Corp.	85
328	Texas Computer Systems	181
65	Tora Systems Limited	190
95	Total Information Services	41
•	Trionyx Electronics	189
263	US Robotics	70
221	US Robotics	98
181	US Robotics	181
•	VR Data Corporation	151
214	VanHorn Office Supply	103
•	Wameco	169
163	Wintek Corporation	193
122	World Wide Electronics	64

*This advertiser prefers to be contacted directly.

Reader Responsibility

One of your responsibilities, as a reader of *Kilobaud Microcomputing*, is to help increase the magazine's circulation and advertising, both of which will bring you the same benefit—a larger and even better magazine. You can help by encouraging your friends to subscribe to *Kilobaud Microcomputing*. Remember: Subscriptions are guaranteed—money back if not delighted. So no one can lose. You can also help by filling out the Reader Service card. Answer the questions in the questionnaire and circle the advertisers' Reader Service numbers for more product information. Advertisers put a lot of trust in readers' requests for information.

kilobaud MICROCOMPUTING

For further information from our advertisers, please use the Reader Service card.

READER SERVICE

Reader Service: Return this card to receive full information on the products advertised in this issue. Refer to the ad. You will find numbers near the logo of each advertiser. Each represents the advertiser's individual Reader Service number. Circle the corresponding numbers on one of the cards on this page, include your name, address & zip, and drop in a mailbox. In 4-6 weeks you'll hear from the advertiser directly.

This card valid until May 31, 1981.

Please help us to bring you a better magazine—by answering these questions:

I. I am interested enough in a general-interest article to see it in lieu of a technical article.
- ☐ 1 Yes
- ☐ 2 No

II. The following types of general interest articles appeal to me most (check two only).
- ☐ A Humor
- ☐ B Interview with users/industry leaders
- ☐ C Historical
- ☐ D Essay
- ☐ E Profile of a business

III. As a reader of Microcomputing, I consider myself a
- ☐ 1 Beginner
- ☐ 2 Intermediate
- ☐ 3 Expert

IV. The articles in Microcomputing are
- ☐ A Too simple
- ☐ B Too complex
- ☐ C Just right

V. I want to see more industry news in Microcomputing
- ☐ 1 Yes
- ☐ 2 No

VI. The following major computer publication gives me the most usable information (check only one)
- A Kilobaud Microcomputing
- B 80 Microcomputing
- C Byte
- D Interface Age
- E Personal Computing
- F Creative Computing
- G Dr Dobb's Journal

VII. I want to see more regular departments in Microcomputing
- 1 Yes
- 2 No

VIII. I bought this magazine on the newsstand because (answer only if question applies)
- A I was looking specifically for Microcomputing
- B The cover caught my eye

IX. I want to see more of the following kinds of articles in Microcomputing (check two only)
- 1 Business
- 2 Construction
- 3 Education
- 4 Game
- 5 Home application
- 6 Human interest
- 7 Programming technique
- 8 Review
- 9 Scientific

X. Is the advertising content of the magazine a primary reason for your buying Kilobaud?
- A Yes
- B No

XI. What type of computer do you own? (Check all that apply)
- 1 Apple
- 2 PET
- 3 TRS-80
- 4 Atari
- 5 Heath
- 6 TI
- 7 Other

XII. If you are not a subscriber, please circle 500.

Kilobaud MICROCOMPUTING • March 1981

1	6	11	16	21	126	131	136	141	146	251	256	261	266	271	376	381	386	391	396
2	7	12	17	22	127	132	137	142	147	252	257	262	267	272	377	382	387	392	397
3	8	13	18	23	128	133	138	143	148	253	258	263	268	273	378	383	388	393	398
4	9	14	19	24	129	134	139	144	149	254	259	264	269	274	379	384	389	394	399
5	10	15	20	25	130	135	140	145	150	255	260	265	270	275	380	385	390	395	400
26	31	36	41	46	151	156	161	166	171	276	281	286	291	296	401	406	411	416	421
27	32	37	42	47	152	157	162	167	172	277	282	287	292	297	402	407	412	417	422
28	33	38	43	48	153	158	163	168	173	278	283	288	293	298	403	408	413	418	423
29	34	39	44	49	154	159	164	169	174	279	284	289	294	299	404	409	414	419	424
30	35	40	45	50	155	160	165	170	175	280	285	290	295	300	405	410	415	420	425
51	56	61	66	71	176	181	186	191	196	301	306	311	316	321	426	431	436	441	446
52	57	62	67	72	177	182	187	192	197	302	307	312	317	322	427	432	437	442	447
53	58	63	68	73	178	183	188	193	198	303	308	313	318	323	428	433	438	443	448
54	59	64	69	74	179	184	189	194	199	304	309	314	319	324	429	434	439	444	449
55	60	65	70	75	180	185	190	195	200	305	310	315	320	325	430	435	440	445	450
76	81	86	91	96	201	206	211	216	221	326	331	336	341	346	451	456	461	466	471
77	82	87	92	97	202	207	212	217	222	327	332	337	342	347	452	457	462	467	472
78	83	88	93	98	203	208	213	218	223	328	333	338	343	348	453	458	463	468	473
79	84	89	94	99	204	209	214	219	224	329	334	339	344	349	454	459	464	469	474
80	85	90	95	100	205	210	215	220	225	330	335	340	345	350	455	460	465	470	475
101	106	111	116	121	226	231	236	241	246	351	356	361	366	371	476	481	486	491	496
102	107	112	117	122	227	232	237	242	247	352	357	362	367	372	477	482	487	492	497
103	108	113	118	123	228	233	238	243	248	353	358	363	368	373	478	483	488	493	498
104	109	114	119	124	229	234	239	244	249	354	359	364	369	374	479	484	489	494	499
105	110	115	120	125	230	235	240	245	250	355	360	365	370	375	480	485	490	495	500

Name ______________________________

Address ______________________________

City ______________________ State ________ Zip ________

Microcomputing • POB 2741 • Clinton IA 52735

BOOKS

Please send me the following Microcomputing products:

Qty.	Catalog #	Title	Unit Price	Total

Add $1 shipping/handling ________

Total ________

Please allow 4–6 weeks for delivery. No C.O.D.s accepted.

Enclosed $____________ ☐ Check ☐ M.O.

Bill: ☐ AE ☐ MC ☐ Visa

Card # ______________________ Exp. date ________

Signature ______________ Interbank # ____________

Name ______________________________

Address ______________________________

City ______________________________

State ______________________ Zip ________

3/81

Microcomputing • Mail Order Dept. • Peterborough NH 03458

SUBSCRIPTION

MICROCOMPUTING subscribers save $10 off the newsstand price.

☐ New subscription ☐ Renewal

☐ 1 year —$25

☐ 2 years—$38

☐ 3 years—$53

Enclosed $____________ ☐ Check ☐ M.O.

Bill: ☐ MC ☐ Visa ☐ AE ☐ me

Card # ______________________ Exp. date ________

Signature ______________ Interbank # ____________

Name ______________________________

Address ______________________________

City ______________ State ____________ Zip ________

Canada—$27, 1 year only, US funds

Other foreign—$35, 1 year only, US funds

313B9

Microcomputing • POB 997 • Farmingdale NY 11737

kilobaud MICROCOMPUTING

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 217 CLINTON IA 52735

POSTAGE WILL BE PAID BY ADDRESSEE

kilobaud
MICROCOMPUTING T.M.

POB 2741
Clinton IA 52735

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 1024 PETERBOROUGH NH 03458

POSTAGE WILL BE PAID BY ADDRESSEE

kilobaud
MICROCOMPUTING T.M.

Subscription Dept.
POB 997
Farmingdale NY 11737

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 1024 PETERBOROUGH NH 03458

POSTAGE WILL BE PAID BY ADDRESSEE

kilobaud
MICROCOMPUTING T.M.

Peterborough NH 03458

Att. Mail Order

✓ Reader Service—see page 178

Edited by Dennis Brisson

New HP Business Computer
Atari Expansion
12 V dc Printer
PET Battery Backup

Hewlett-Packard Computer And Business Software

Hewlett-Packard, 1507 Page Mill Road, Palo Alto, CA 94304, has introduced a new microcomputer and new software for business applications.

The HP-83 comes with an integrated high-resolution CRT and keyboard, enhanced BASIC language and graphics capabilities. Designed for business and technical professionals, it is identical to the HP-85 except that it does not come with an integrated tape cartridge drive and integrated thermal printer. Price is $2250.

The business software includes the Information Management Pac, which provides a data base management tool for applications such as list keeping, inventory and reporting, and the Graphics Presentation Pac for producing charts and text for reports and overhead projection transparencies. The new software runs with both the HP-83 and HP-85. Reader Service number 487.

Hewlett-Packard's HP-83.

Daisywheel Printer

The Starwriter Daisywheel printer runs at 25 cps and features a Centronics-compatible parallel interface. This letter-quality receive-only printer uses Diablo ribbons and printwheels. It has full graphics capability (1/120-inch horizontal control, 1/48-inch vertical control) and is completely code compatible with Qume and Diablo, so no software changes are required to use boldface, underlining and bidirectional printing. It accommodates paper widths of up to 15 inches and will furnish three clear copies. Price is $1779.

Computer Textile, Inc., 10960 Wilshire Blvd., Suite 1504, Los Angeles, CA 90024. Reader Service number 493.

S-100 Error-Correcting Board

An S-100 error-correcting board for error correction or system problem diagnosis is available from Correlation Systems, 81 Rockinghorse Road, Rancho Palos Verdes, CA 90274. It monitors the presently existing system RAM via the bus signals, intervening to correct erroneous bus data before, it's accepted by the CPU. The board corrects all one-bit memory errors and flags all one-bit or two-bit errors, increasing system memory reliability. All operations are performed through on-board hardware. Latched displays show the address and bit in error. Operating to 4 MHz, the board protects 64K of memory. Price is $1295. Reader Service number 492.

Atari 32K RAM Expansion

A new memory expansion system from Axlon, Inc., 170 N. Wolf Rd., Sunnyvale, CA 94086, upgrades the Atari 400 to 32K and the Atari 800 to 48K RAM. RAMCRAM contains 16 memory chips, yielding a total of 32K of additional user program memory. It is installed in the Atari 400 by removing the top enclosure of the computer console and unplugging the 8K RAM module supplied by Atari. The RAMCRAM module is then plugged into the same slot. This allows you to plug in disk drives, printers and any other peripheral devices formerly compatible only with the Atari 800. Any 32K Atari 800 software on the market will run on a 400 with RAMCRAM. For the 800, putting 32K into one memory slot allows upgrading of the system to 48K with one entire slot left over for future expansion. Reader Service number 495.

North Star Floppy Subsystem

An eight-inch floppy subsystem that allows North Star users greater auxiliary storage and the ability to transfer files from 5-1/4-inch media to eight-inch media or vice versa is available from John D. Owens Associates, Inc., 12 Schubert St., Staten Island, NY 10305. This fully integrated hardware/software package is designed to operate in single or double density on eight-inch media in IBM-compatible format.

The hardware, which requires simple, plug-in installation and is completely compatible with North Star hardware, includes a Tarbell double-density controller, dual Shugart 800R drives housed in a cabinet with power supply, fan and all necessary cabling.

The software interface, DMA-DOS, is a high-speed, single-user CP/M-compatible 8080/Z-80 disk operating system that features easy-to-use commands and utilities that

allow extensive control over processing activity. Price is $1910. Reader Service number 490.

Syntest's dot matrix printer.

Printer for 12 V dc Operation

For the computerist on the go, Syntest Corporation, 169 Millhan St., Marlboro, MA 01752, offers the SP-314 dot matrix alphanumeric printer that operates on 12 V dc power, thus allowing complete independence from conventional power sources. This unit features buffered 40-column impact printing on 4.25-inch paper, RS-232 serial and parallel inputs, crystal-controlled baud rate, 96-character ASCII set plus double width and a self-test routine. This 8.5 × 5.75 × 8.5 inch unit weighs 5.5 lbs. and includes graphics capabilities. Price is $605. Reader Service number 498.

Short-Length Cassettes

Microsette Co., one of the first companies to recognize the need for short-length cassettes in the microcomputer market, has added the C-60 and the C-90 length cassettes to its product line, which features the C-10 (50 ft.) and the C-20 (100 ft.) lengths already available. The C-60 measures 290 ft., and the C-90 is 435 ft.

Microsette Co., 475 Ellis St., Mountain View, CA 94043. Reader Service number 488.

Battery Backup for PET

Battery backup systems for the Commodore PET and Dual Disk Floppy Drive are available from ETC Corporation, PO Box G, Apex, NC 27502.

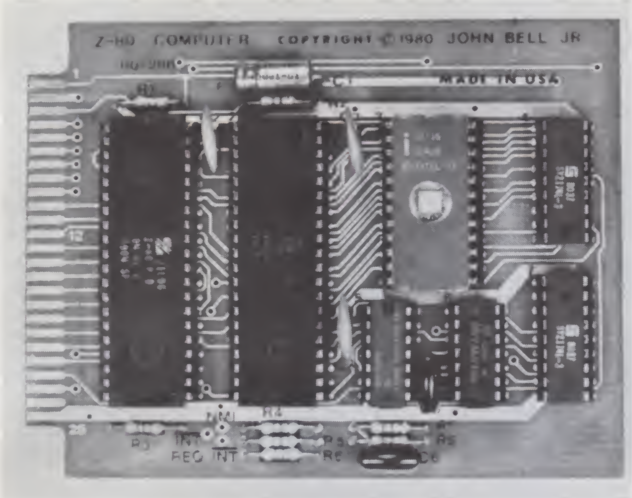
Backpack, designed to install within the CPU case, gives six to ten minutes of full power (CRT included) to the PET during power outages and glitches in power lines, allowing you time to save data or programs to disk or tape. Batteries are recharged from the PET's own internal power supply. Price is $225.

Floppy Backpack, used in conjunction with Backpack, gives 15 to 20 minutes of reserve power for both of Commodore's floppy drives. It allows you to save data and programs during power outages and also eliminates possible disk crashes. Designed to install within the disk cabinet with no wiring changes, the batteries are recharged from the disk internal power supply. Price is $135. Reader Service number 494.

Z-80 Single-Board Computer

John Bell Engineering's Z-80-based single-board dedi-

Microsette's cassettes.



John Bell's Z-80 microcomputer.

cated computer, designed for control functions, is software-compatible with the Z-80, 8080 and 8085 microprocessors. This Z-80 microcomputer uses a Z-80 PIO chip that has two independent eight-bit bidirectional peripheral interface ports with handshake and data transfer control. It can be interrupt-driven. It uses one 2716 for EPROM memory and two 2114 RAMs and requires a single 5V power supply at 300 mA. Prices are $129.95 (assembled and tested), $119.95 (kit) and $29.95 (bare board).

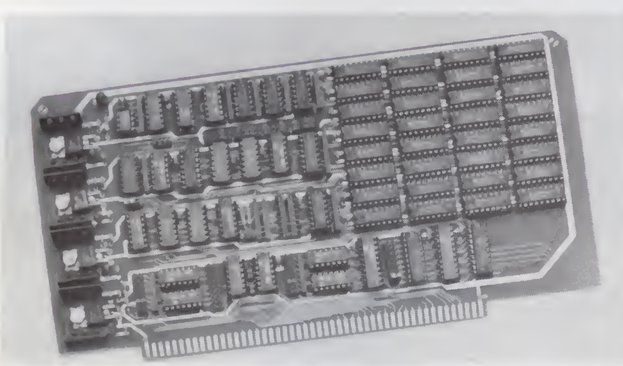
John Bell Engineering, PO Box 338, Redwood City, CA 94064. Reader Service number 489.

8/16-Bit Memory Board

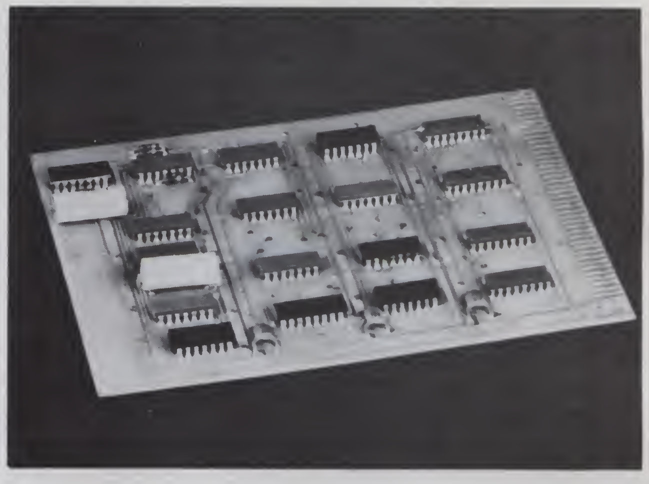
The MB10, from SSM Microcomputer Products, 2190 Paragon Drive, San Jose, CA 95131, is a 4 MHz 16K static RAM memory board that features extended addressing and IEEE-standard protocol. Extended addressing allows the use of multiple boards in the same configuration with as much as 16 megabytes of memory. For those applications requiring bank switching, the MB10 is jumper selectable to any I/O address with enable/disable on power-up or reset (Cromemco-compatible). Automatic 16-bit request and acknowledge protocol allows mixing 16-bit and eight-bit microprocessors in the same mainframe. It supports the new SSM CB2 Z-80 CPU board and will also operate an 8K-word RAM board for future 16-bit microprocessors. Price is $399. Reader Service number 496.

Panel Meter-to-RS-232C Converter

The Model DPT-415 is a digital panel meter-to-RS-232C converter module that accepts parallel inputs and gen-



SSM's 16K RAM board.



Digital Laboratories DPT-415.

erates serial outputs in a computer- or printer-compatible format. When used with a printer, it automatically counts readings and inserts extra formatting characters as required. It can replace a more costly dedicated microcomputer with associated PROM, UART and input multiplexing.

It features simple jumper-type connections that allow the module to be interfaced with a computer or directly to a printer having up to five readings per line. Operating on less than 2 watts, the 6 1/2 inch × 4 1/2 inch unit also has 20 mA current loop provisions to interface with Teletypes. Price is $295.

Digital Laboratories, Inc., 600 Pleasant St., Watertown, MA 02172. Reader Service number 491.

LOWEST PRICE - BEST QUALITY

NORTH STAR

North Star Horizon 2

2-5¼ Disk Drives
32K Double Den
Factory assem. & tested
Factory guaranteed
List 3095

only **$2274**

POWERFUL NORTH STAR BASIC FREE
SUPERB FOR BUSINESS & SCIENCE

FACTORY ASSEMBLED & TESTED	LIST	ONLY
HORIZON-1-32K-DOUBLE DEN	$2695	$1980
HORIZON-2-32K-QUAD DENSITY	3595	2674
HORIZON-2-64K-QUAD + HARD DISK	9329	7149
HORIZON RAM ASSM	16K = $389.	32K = $579
HORIZON RAM KIT	SALE! 16K = $314	32K = $469
HORIZON DISK DRIVE SALE DOUB DEN	SAVE!	315
NORTH STAR HARD DISK 18 Mb	4999	$3929
PASCAL-PLUS 14,18 OR 36 DIGIT PRECISION		249

SUPERBRAIN

ZENITH

SUPERBRAIN QD 64K
List $3995 only $2995

Z-89 48K
List $2895 only $2299

TERMINALS Z-19 $725

INTERTUBE III only **$725**

DIP-81 PRINTER only **$395**

MICROTEK $675

NEC PRINTER $2569

TRACTOR, THIMBLE, RIBBON

InterSystems

ITHACA INTERSYSTEMS 2A

Z-80A CPU 4 MHz
64K Dynamic RAM
Front panel
V I/O—with interrupts
FDCII Disk Controller
20 slot motherboard

List $3595 You Pay Only **$2795**

PASCAL/Z + THE FASTEST PASCAL $375
GET READY FOR ITHACA'S Z-8000
8086 16 BIT CPU & SUPPORT CARD SEATTLE $575

MORROW 8" DISK

DISCUS 2D + CP/M® 600K ONLY $938
DISCUS 2 + 2 + CP/M® 1.2 MEGA B. $1259
ADD DRIVES 2D = $650 2 + 2 = $975
2D-DUAL + CP/M® + MICROSOFT BASIC = $1555
$2293 VALUE OVER 32% OFF!
WHILE THEY LAST

MORROW HARD DISK
26,000,000 BYTES!!
LIST $4995 ONLY $3995
CP/M® IS INCLUDED!

SAVE ON MEMORY AND PROGRAMS

SYSTEMS MEMORY 64K A & T 4mHz	$599
SYSTEMS MEMORY 64K BANK SELECT	$789
CENTRAL DATA 64K RAM	$599
ITHACA MEMORY 8/16-bit 64K	$845
SEATTLE MEMORY 8/16 BIT 16K 4Mhz	$275
SSM KITS Z-80 CPU	$221
VIDEO BRD V83 4Mhz	$412
ANADEX PRINTER DP-9500-1	$1349
CAT NOVATION MODEM	$169
TARBELL DISK CONTROLLER DD	$445
ECONORAM 2A 8K ASSM	$179
NSSE 1-22 & P01 TERRIFIC PROGRAMS	ONLY $10. EACH
NORTHWORD 294 MAILMAN 234 INFOMAN	$364
TARBELL COMPUTER-PHONE	
RCA-COSMAC VP-111 99 RCA-COSMAC VP-711	$199
COLOR! RAINBOW 385 SPECTRUM	$289
EZ-80 Machine Language Tutor	$25
EZ-CODER Translates English to BASIC	$71
ECOSOFT FULL ACCOUNTING PKG	$315
BOX OF DISKETTES	$29
SECRETARY WORD PROCESSOR	The Best! $99
TEXTWRITER III Book Writing Program	$112
GOFAST NORTH STAR BASIC Speeder Upper	$71
Which Computers are BEST? BROCHURE	FREE
North Star Documentation refundable w/HRZ	$20

AMERICAN SQUARE COMPUTERS BEATS ADV. PRICES

American square Computers ✓56

919-889-4577 KIVETT DR. JAMESTOWN N.C. 27282 919-883-1105

® CP/M is a registered trademark of Digital Research, Inc.

Edited by Dennis Brisson

"Talking" Game and Business Programs
Apple II Business, Color Graphics
Office-Management Package
Accounting, Tax-Planning Software

Speech Driver and Speech Games

Percom Data Company, 211 N. Kirby, Garland, TX 75042, is now offering an advanced speech driver and eight "talking" game programs for their Speak-2-Me-2 interface. The Speak-2-Me-2 printed circuit card installs in a Texas Instruments' Speak & Spell and, by providing the interface between the Speak & Spell and a computer, allows you to add speech to business and game programs. The Speak-2-Me-2 module and software are configured for TRS-80 Models I or III, but may be adapted for other computers.

The driver program provides a direct word output routine for outputting words and phrases of the Speak & Spell vocabulary and includes routines which will create new words using syllables of Speak & Spell words and a routine which will modulate the speed at which words are spoken. The game programs are speech-enhanced by vocal prompts and responses uttered at appropriate points during play. Included among the games are Hangman, card and dice games of chance and a challenging version of Simon Says. The driver routines and game programs are supplied on diskette for $29.95. The Speak-2-Me-2 adapter sells for $69.95. Reader Service number 478.

Ultrasort II

Ultrasort II gives CP/M and CBASIC2 users a fast 8080/Z-80 machine-language program to sort, merge or select records from data files, or to find the number of logical records in a file. It can be used as a stand-alone utility or as a fast subroutine called from CBASIC2. This machine-language program handles large data files rapidly and will sort records thousands of bytes long. It sorts on up to five keys, each independently ascending or descending.

Fields may be variable or fixed length. The select capability lets you either omit or include records that are less than, equal to or greater than up to four independent select keys. The program also provides prompted disk changes for work and output files.

Computer Control Systems, Inc., 298 21st Terrace S.E., Largo, FL 33541. Reader Service number 475.

Tax-Planning Program

Shortax is a tax-planning program that enables professionals to forecast and analyze federal and social security taxes in just minutes, instead of hours. The program calculates federal income tax (using tax rate tables, income averaging formula, optional maximum tax method and the corporate alternative tax calculation for long-term capital gains), the add-on minimum tax, the alternative minimum tax, the FICA tax and the self-employment tax. It also forecasts positive or negative compound growth rates or constant incremental amounts for up to 99 periods. Data files can be stored, retrieved, recalculated, modified, listed, added together, netted against each other or deleted with menu-prompting functions.

Data input, which is cross-referenced to tax forms with on-line instructions, consists of just 20 variables. Detailed instructions are available in response to "?" or "help" input. The program runs on most microcomputer systems which use a Micropolis Disk Operating System and a CPU with at least 48K of memory; it will also run on a Radio Shack Model I, Level II with 48K of CPU memory and one (or more) Radio Shack disk drives. Program price is $500.

Syntax Corp., 4500 W. 72nd Terrace, Prairie Village, KS 66208. Reader Service number 481.

Z-80 Accounting Software

The Software Fitness Program, written in COBOL and run under the OASIS operating system, is a set of financial accounting applications for Z-80-based microcomputers. It is designed for small-business users and includes Sales Order Processing, Accounts Receivable with Billing and Sales Analysis, Accounts Payable, General Ledger, Inventory, Payroll and Job Cost. It will run on single- or multiple-user hardware systems supporting both disk and diskette storage. Software features include open item and balance forward accounting; LIFO, FIFO and average cost methods of inventory valuation; single or multiple company processing; revenue and expense accounting by job or phase; and a report generator for financial statements. Interface switches can be set by the user to control the complexity of accounting interactions.

Open Systems, Suite 409, 430 Oak Grove, Minneapolis, MN 55403. Reader Service number 484.

Professional Software

Professional software packages to aid the dentist, attorney and consultant in appointment scheduling, professional time management, private client billing and management reporting has recently been released by CompuSo-Co, 26251 Via Roble, PO Box 2325, Mission Viejo, CA 92690.

Professional I—for the dental professional—features preparation for ADA claim forms for third-party patients and notices for dental checkups.

Professional II—geared to the needs of the legal professional—features preparation of special reports for third-party legal plans and special accounting plans to analyze court time usage and work on retainer or contingency engagements.

Professional III—a general-purpose package for consultants, accountants and contract administrators—allows the creation of sub-jobs, special cost centers, overhead accounts, billing under time and materials contracts, fixed-priced job cost accounting

$ DISCOUNT $

TRS-80® DEALER A301

COMPUTER SPECIALISTS

MODEL II

26-4002
64K 1 Drive
$3440.00

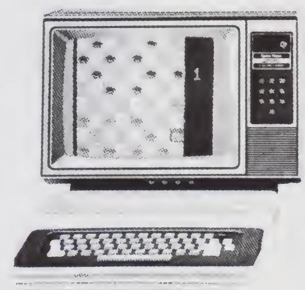
26-4160 1 Drive EXP.	$1035.00
26-4161 2 Drive EXP.	1575.00
26-4162 3 Drive EXP.	2115.00
26-4501 Gen. Ledger	180.00
26-4502 Inventory	180.00
26-4503 Payroll	360.00
26-4554 Acct. Rec.	180.00
26-4701 Fortran	270.00
26-1157A Daisy Wheel	2290.00
26-1158 Daisy Wheel II	1799.00

26-1155 Quick Printer II	$187.00
26-1145 RS-232 Board	84.00
26-1140 "O" K Interface	249.00
26-1141 "16" K Interface	359.00
26-1142 "32" K Interface	469.00
26-1160 Mini Disk - Drive O	419.00
26-1161 Mini Disk - Additional	419.00
26-1154 Lineprinter II	699.00
26-1165 Line Printer V	1710.00
26-1159 Lineprinter IV	859.00
26-1166 Line Printer VI	1080.00
26-1563 Scripsit - Disk	79.00
26-1566 Visicalc	83.00
26-1562 Profile	72.00

MODEL III

26-1061 4K I	$630.00
26-1062 16K III	888.00
26-1063 32K III	
2-Drives, RS232	2225.00

COLOR



26-3001 4K	$360.00
26-3002 16K	540.00
26-3010 Color Video	360.00
26-1206 Recorder	54.00
26-3008 Joysticks	22.50

CENTRONICS

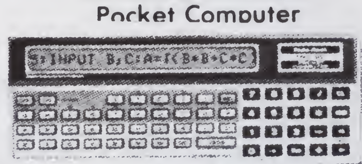
Fast 100 CPS Centronics 730 Printer	577.00
Text Quality Centronics 737 Printer	737.00

Model II Cobol Compiler
$360.00
Cobol Run Time Package
$36.00

ALL OTHER R.S. SOFTWARE FURNITURE, STANDS, CABLES AND ACCESSORIES AT DISCOUNT FROM CATALOG PRICE.

Novation Cat Modem	$149.00
CCA Data Management System	72.00
Adventure Games Games 1-9 each	14.00

Pocket Computer



26-3501 1.9K P.C.	$225.00
26-3503 Cassette I/F	45.00
14-812 Recorder	72.00

Acorn Software Products, Inc.

GAMES:	
Alien Invasion	$9.00
Stock Market	9.00
Star Trek	9.00
Block Em	9.00
Ting-Tong	9.00
UTILITIES:	
System Savers	14.00
EDUCATION:	
Language Teacher	18.00

FREE: COMPUTER CATALOG UPON REQUEST

1-800-841-0860 Toll Free Order Entry

MICRO MANAGEMENT SYSTEMS, INC. ✓100

No Taxes on Out Of State Shipments

Immediate Shipment From Stock on Most Items

DOWNTOWN PLAZA SHOPPING CENTER
115 C SECOND AVE. S.W.
CAIRO, GEORGIA 31728
(912) 377-7120 Ga. Phone No.

R.S. 90 Day Limited Warranty
F-48 Form Provided

Largest Inventory
In the S.E. U.S.A.

*TRS-80 is a registered trademark of the Tandy Corp.

✓ Reader Service—see page 178

and many other job set-up systems.

All systems require an Apple II or Apple II Plus computer with Applesoft, a 130-column printer and at least two minifloppy disk drives. The system will handle up to 150 professional practitioners with client bases of up to 10,000 clients each. Price is $750. Reader Service number 486.

Paintset for the Apple

Micro-Painter uses high-resolution graphics to "paint" pictures in 21 different colors on the Apple II.

Designed for computer hobbyists and professional programmers, it can be used in personal and business applications for presentations, demonstrations, computer-assisted instruction and home entertainment.

It includes a magnification feature for dot-by-dot coloring and inverse coloring. Once painted, pictures can be saved or displayed in any combination of colors or in an unpainted state. Pictures can also be repainted at any time. Price is $34.95.

Datasoft, 16606 Schoenborn St., Sepulveda, CA 91343. Reader Service number 471.

Intelligent Terminal Emulator

Cyber-Smart is an intelligent terminal program designed to communicate with mainframe computers and with other computers that have an RS-232 port. It features a fundamental set of communication routines capable of remote intelligent terminal emulation. The program can also be configured in a master/slave mode for computer-to-computer interaction.

The program is designed for expansion and personal modification in either BASIC or machine language. It uses less than 4K of RAM and can be configured for 8K, 16K, 24K or 32K systems. It is available for the OSI cassette-based C4P computer. Price is $29.95. An RS-232 port installation kit is available for $14.95. The combined software/hardware kit costs $39.95.

Cyber Associates, Inc., PO Box 4187, Dept. M, Huntsville, AL 35802. Reader Service number 474.

Atari Graphics Editor

Plot & Draw is a graphics package for the Atari that allows you to readily generate graphics in three colors, plus a background. You can create intricate video drawings and save them on cassette tape, where they can be retrieved later by your own computer programs. Single keystroke commands and the ability to combine pictures allow you to produce complex drawings. The program continuously displays its cross-hair coordinates for precision and flexibility. It comes on 8K cassette for Atari computers equipped with a joystick. Price is $18.

Mosaic Electronics, Box 748, Oregon City, OR 97045. Reader Service number 472.

TRS-80 Income Property Analysis

Incoprop is an income property analysis program for the TRS-80 Models I or III. Designed for real-estate brokers, sales people, counselors, educators, students and investors, the program is based upon standardized methods of property analysis used throughout the real-estate industry. It generates a property operating statement and a ten-year cash flow analysis. The use of a built-in IRS tax table ensures high accuracy of an investor's projected after-tax income stream. Minimum system requirements include 32K RAM, one disk drive and a printer. Price is $120.

E-Z Software, PO Box 591, Novato, CA 94947. Reader Service number 482.

TRS-80 Accounts Receivable

AR, an accounts receivable system for the TRSDOS 1.2 on TRS-80 Model II, is a complete invoicing and monthly-statement-generating system which keeps track of current and aged accounts receivable.

AR enables you to speed collections and spot potential losses early enough to take action. This package maintains a file consisting of the customer's name, address and phone number, along with the customer's type of account (open item, balance forward or auto billing), current balance, tax rate and other useful account status information. It uses a full 80-column screen and requires 64K memory and dual disk systems. Price for the eight-inch diskette program is $129.

Micro Architect, Inc., 96 Dothan St., Arlington, MA 02174. Reader Service number 476.

6502 Disk Operating System

DOS/65 is an operating system for the 6502 family of processors. It uses a layered structure like CP/M so that each user's peculiar hardware and software environment can be accommodated with minimal effort. System requirements include an eight-inch IBM-compatible drive and controller, a console input device and a console output device, as well as a minimum of 16K to 24K RAM, depending upon the host system. In addition to the basic operating system, the diskette contains a disk file text editor, a disk-based two-pass assembler, a debugger, a system generation routine and other transient utilities.

Richard A. Leary, 1363 Nathan Hale Drive, Phoenixville, PA 19460. Reader Service number 483.

North Star Z-80 BASIC

Now the North Star Horizon can execute programs 25–30 percent faster with Bazic, a software package with all the same functions, as well as several additional features, provided by North Star BASIC. Written in Z-80 code, Bazic features increased speed and more compact code, requiring less memory than BASIC. Bazic simplifies programming with the use of APPEND as a command or as an executable statement in a program. It also provides functions to assist in screen formatting. The Bazic version using North Star DOS as the disk operating system is available on 5-1/4-inch disk for North Star double-density/quad-capacity systems for $150.

Micro Mike's, Inc., 905 South Buchanan, Amarillo, TX 79101. Reader Service 479.

Instructor Gradebook for the Apple II

Now educators at every grade level can record and report individual and class performance for classes of up to 400 students and statistically measure the effectiveness of teaching and evaluation techniques with the Instructor Gradebook from Serendipity Systems, Inc., 225 Elmira Road, Ithaca, NY 14850. Designed for the Apple II Plus or the Apple II with Applesoft firmware, Instructor Gradebook offers seven defined and two optional grading categories (e.g., test, homework, quiz). Marks can be entered either as numbers or as letter grades. Reports available include alphabetized class lists, grade listings by student ID numbers for posting individual performance reports and permanent class records. Price is $169. Reader Service number 477.

Business Graphics For the Apple

Trend-Spotter is an easy-to-use business graphics and analysis for the Apple II Plus computer. It allows you to spot emerging trends through quick manipulation and graphing of business data. It will generate and read VisiCalc-compatible files. Price is $275.

Software Resources, Inc., 44 Brattle Street, Cambridge, MA 02138. Reader Service number 473.

✓ Reader Service—see page 178

COMPUTER CLINIC

I have recently changed my CPU computer board from an 8080A to a Z-80 and now find that my Altair Extended BASIC will not run. Does anyone know of a "patch" to Altair BASIC which will allow it to run on the Z-80?

George R. Steber
9957 N. River Road
Mequon, WI 53092

I need schematics and service data for a Tektronix 31 calculator and Tektronix 4661 digital plotter. Attempts to get this information from Tektronix have been unsuccessful.

Kenneth H. Reid
1935 Trevilian Way
Louisville, KY 40205

I need a BASIC interpreter (preferably 8K or larger) on Tarbell cassette suitable for use with an Imsai 8080. I also want the source code listing, if available. Since I live in Saudi Arabia, where computing materials are unobtainable, any help would be much appreciated.

Richard Herman
c/o Donald Herman
Thor Johnson Dorm
Interlochen Arts Academy
Interlochen, MI 49643

I have acquired two eight-inch single-sided disk drives, which, I believe, might be IBM and are marked "33FD," but I have no data on them. I would be grateful if any reader of *Microcomputing* could assist me with any relevant data—particularly details of the pinouts from the 24-pin connectors on the side of them.

Rory O'Farrell
Tinode
Blessington
Co. Wicklow
Ireland

CLUB NOTES

Hamilton, Ontario

The OSI User's Group of Southern Ontario will hold its next meeting on Saturday, March 7, from 10 a.m. to 4 p.m., at McMaster University, General Sciences Building, Room 312. Meeting dates for the remainder of 1981 are June 6, September 5 and December 5. For more information, contact Dr. N. Solntseff or C. Bryce, Unit for Computer Science, McMaster University, Hamilton, Ontario L8S 4K1, 416/525-9140, ext. 4689 or 2065.

Delaware Valley

Computer owners of the Delaware Valley should note the existence of the TRS-80 User's Group of Delaware. Contact Tim, 302/478-7415; Merle, 302/366-0785; or Al, 302/762-4223.

Rochester, NY

The Rochester Area Microcomputer Society will hold its third annual spring computer show on April 4 at the Perinton Square Mall, corner of Rts. 250 and 31, Fairport, NY, near Rochester. For further information, contact the Rochester Area Microcomputer Society, PO Box 90808, Rochester, NY 14609, or call Erwin Rahn, president, 716/473-3184.

New Orleans, LA

The Texas Instruments Minicomputer Information Exchange (TI-MIX) will hold a conference for users of Texas Instruments computer equipment on March 8–11 at the Marriott Hotel in New Orleans. Further information is available from TI-MIX, PO Box 2909, Austin, TX 78769, 512/250-7151.

Son of Cheap Video
Interactive Computer Graphics
Interfacing with the 8255 PPI

Son of Cheap Video

Don Lancaster
Howard W. Sams & Co., Inc.
$8.95

Don Lancaster is a capitalist's capitalist. He fully understands the concept that to make money, you must find ways to not spend money. The result of Lancaster's capitalism has been a series of "you-build-it" hardware books. The latest in the series is *Son of Cheap Video*, a sequel to *The Cheap Video Cookbook*, a volume of video hardware fixes and modifications for the hardware freak.

In *The Incredible Secret Money Machine*, Lancaster asserts that the secret money machine is you, and your willingness to put your micro to work either by writing programs, taking care of the statistics for a local bowling league or designing hardware modifications that can make a microcomputer do more for less. In *Son of Cheap Video*, he is, once again, practicing what he preaches.

The concept of cheap video is simple—find new ways to get the output of a microcomputer onto an ordinary TV screen. This is done with a minimum of hardware and without major modifications to either the computer or the TV. Lancaster calls this concept "scungy video."

The advantages of scungy video, designed primarily for the KIM-1 and other 6502-based systems, are many. You can eliminate most of the address restrictions that cheap video imposes on your micro; scungy video techniques eliminate much of the interaction between the computer and the video circuitry, making the video hardware a mere add-on to the system; and you can eliminate one or both of the custom PROM memories used in cheap video.

Scungy video works by adding two new elements to the software scan microinstruction and hardware upstream tap that are the keys to cheap video. It eliminates the subroutine address space mapping used in cheap video and uses break mapping for the scan microinstruction instead. Also, scungy video uses available parallel I/O ports to decrease special hardware needs.

With cheap video and scungy video, complex systems timing and separate display memory are not needed, because the system lets the computer do the work on a time-sharing basis with the programs in memory. The net result of these modifications is a video display system that can be built for as little as $7.

And that's just the beginning. *Son of Cheap Video* will also show you how to achieve full transparency, which lets you compute and display at the same time. All it takes is a length of wire (used to make a sensor coil) and an extra CMOS gate. This new route to transparent cheap video is called the "snuffler" method. Or, if you need to deliver a message in Swahili, using Icelandic characters, *Son of Cheap Video* will show you how to generate custom characters and graphics. As an example of what you can do with the character generator, chapter 4 shows you how to put together a music display with a music staff; whole, half, quarter, eighth and sixteenth notes; measure symbols; treble and bass clefs and more.

Son of Cheap Video doesn't forget the 8080/Z-80 crowd, either. There are adaptations of cheap video for the Heath H8 on the hardware and software sides. In a similar vein, there are two chapters detailing a hardware/software modification to add lowercase capability to the Apple II. This lowercase modification uses the existing keyboard, and you add only two new ICs and a plug-in module. No foil cuts or PC board changes are necessary. Lowercase for the Apple is the crowning touch to this volume of high-tech fixes for the low-budget fixer and, by itself, justifies the purchase price of the book.

The book is profusely illustrated and contains the kinds of circuit diagrams that almost anyone can read. The information is accessible, and Lancaster's breezy style is often good for a chuckle or two. There are armfuls of information packed on every page.

Don Lancaster knows that tomorrow's hardware innovators are today's low-budget hobbyists, and he has designed this and his other books to teach and encourage that audience of hardware bashers. As Lancaster warns in his introduction, "If you are not one of us, go away."

G. Michael Vose
Instant Software
Peterborough, NH

Principles of Interactive Computer Graphics

Second Edition
William M. Newman and Robert F. Sproull
McGraw-Hill, Inc., 1979
$26.95

For anyone wishing to learn about computer graphics, *Principles of Interactive Computer Graphics* is the place. This book gives both comprehensive and detailed coverage of the current state of the art. And for a field as broad as computer graphics, that's an impressive achievement.

The book is broken into six sections: Basic Concepts, Graphics Packages, Interactive Graphics, Raster Graphics, Three-Dimensional Graphics and Graphics Systems. Basic Concepts is a brief overview of the hardware and software involved in displaying pictures on a screen. The organization of this section is typical of the rest of the book: it goes from easy (points and lines) to difficult (transformations, clipping, windowing). Nevertheless, the writing is clear and it is easy to skip around difficult or uninteresting sections without getting lost in the sections ahead.

Graphics Packages is about display files—how they are organized, constructed, compiled and used to represent geometric models and pictures. The hardware on which the discussion is based is often

available to university students, but is older and more expensive than what a home computer user is likely to have. Nevertheless, the importance of organization and structure in the design of graphics systems makes these chapters relevant, regardless of what hardware you are working on.

Interactive Graphics is where things start to get interesting. It starts out with descriptions of graphical input devices, including light pens, joysticks and the SRI mouse (some are available commercially and some are not). Then it introduces the software techniques for handling them and strategies for making effective and comfortable use of them. If you've never seen what kind of programming is necessary to make these devices work, you'll find this section most enlightening.

Raster Graphics is the section of the book most relevant to microcomputer users. It is about the use of the type of graphics hardware most often found on microcomputers, bit-mapped video displays. The descriptions of the hardware operation and the general-purpose algorithms are directly useful for developing graphics software on microcomputers. (All of the algorithms in the book are presented in Pascal.)

Three-Dimensional Graphics contains some of the most difficult reading in the book. There is no way to cope with the problems of translation and rotation of three-dimensional images without using a lot of mathematics. It is not especially difficult mathematics, and this is as good an explication of it as any, but be prepared. (Two technical appendices introduce the necessary concepts of matrices and coordinate techniques, but you can't learn these subjects from appendices.) The last section, Graphics Systems, brings together much of the technical material in the book and discusses the design of graphics hardware and, most importantly, user interfaces.

Together, these six sections form a rich source of information from which to draw while studying computer graphics. The book includes a good index and an extensive bibliography (over 500 entries) to help you find more details on any of the subjects covered in the text.

Be sure to get the second edition. The first edition is ten years old and is somewhat out of date (it has no discussion of raster graphics, for example). Also, it is only in the second edition that all the sample algorithms are presented in Pascal; the first edition used SAIL, which is not widely known.

As with all hardcover technical books, the price is something else. If you are not sure that you need a really technical overview of the field, try a nearby library and look it over first. But if you are serious about computer graphics, this book is a necessity. Consider the price cheap when compared with all of the expensive hardware it will help you make better use of.

Richard Fritzson
Buffalo, NY

Microcomputer Interfacing With the 8255 PPI Chip

P. F. Goldsborough
The Blacksburg Continuing Education Series
Softcover, 217 pp.
Available from Group Technology, Inc., PO Box 87, Check, VA 24072, for $9.95 postpaid.

If you are considering using an LSI programmable I/O chip and want to know more about its application and theory of operation, or if you are currently using an Intel 8255 in your system but want more flexibility from the chip than simply three ports of eight bits each, then this is the book you need.

If your system can be configured to allow experimentation, the book includes 17 experiments which can be executed by an 8080-based system. Complete program listings (in 8080 assembly language) and circuit diagrams are given for each experiment. Accompanying each experiment is a host of questions, as well as tests for specific results, which test the reader's comprehension of the experiment and the results achieved. The chapter-end experiments reinforce material covered. Some familiarity with digital electronics, microcomputer interfacing, TTL circuitry and solderless breadboarding techniques is assumed.

Since knowledge of I/O techniques, whether accumulator or memory-mapped, is essential to understanding how any interface chip works, both methods are reviewed in chapter 1. Goldsborough discusses characteristics of both techniques and the advantages and disadvantages of each. The experiments in chapter 1 familiarize the reader with a bus monitor, single-step circuit for an 8080 system, a method of counting different types of synchronizing pulses and accumulator versus memory-mapped data input.

Chapter 2 presents an overview of the PPI (programmable peripheral interface) and establishes the basis for the final five chapters. The three modes of operation are explained briefly. The book gives a block diagram of the PPI and covers the various interface signals required by the PPI and generated by the 8080. The PPI internal control logic is discussed in a manner which leaves no doubt how each bit in the mode control byte affects the mode configuration of the PPI. The author gives steps for interfacing the PPI to an 8080 for both accumulator I/O and memory-mapped I/O. Numerous flowcharts supplement the software examples, which helps the readers not familiar with the 8080 assembly language.

Mode 0, or "simple I/O," is covered in

Call For Manuscripts

Kilobaud Microcomputing is looking for business articles!

Businessmen in all fields are beginning to take notice of the microcomputer. They are eager to know which computers, peripheral equipment and applications software will let them take full advantage of this new tool. What knowledge do you have to share?

Here are the kinds of articles that we want you to write for us:

- Are you a businessman with a system up and running? We want to know how it works. What were your expectations? Have they been fulfilled? Did you find the software that you wanted? What problems have you had? How did you overcome them? What recommendations do you have for other businessmen?
- We want reviews from a businessman's perspective of specific hardware and software. If you've recently bought a new product and want to tell others how great—or poor—it is, *Microcomputing* will provide you with a forum.
- What programs have you written to meet your specific needs? Perhaps another businessman can use them, too. Even if he can't, your program may serve as a springboard for other ideas.
- Perhaps you aren't using your micro for business, but know a company that is. Trot on down with your pencil and notebook, and find out what they're up to. While they might not have the time to write up their experiences, they might be more than willing to tell somebody else about them. And an outside observer will often be able to see things with a unique and valuable perspective.

Don't worry if you're not a professional writer. That's what we editors are here for. And we'll be more than happy to send you a copy of our writer's guidelines.

Send your manuscripts and correspondence to:
Kilobaud Microcomputing
Pine St.
Peterborough, NH 03458

chapter 3, which deals exclusively with "unconditional" or "asynchronous" data transfer without handshaking. Various methods of configuring ports A, B and C are discussed. Examples throughout the book illustrate why a particular part configuration might be required. A program and timing diagram follows a section on how to configure the PPI for mode 0 I/O. Port C usage is illustrated with a block diagram, a subport allocation table and two programs. References precede the three experiments. A number of tables in the chapter are handy for reference.

Chapter 4 discusses and illustrates in great detail the PPI bit set/reset capability of port C in mode 0 operation. The reader is given an appreciation for the real-world interface capabilities of the 8255 by an application example.

Status-driven handshaking I/O with combined mode 0 and bit set/reset operation is covered in chapter 5. Handshaking is explained in words and diagrams, and contrasted to interrupt-driven systems in the same manner. Flowcharts are used to excellent advantage in explaining implementation of status-driven handshaking I/O. The interfacing example given in this chapter is a paper tape punch and reader. The experiment covers the subject matter in very good detail. The program examples and circuit diagrams are, as always, given in sufficient detail to let the reader wire-up the circuit and execute a working program without having to guess what the author had in mind and why it doesn't work.

Befitting the subject matter, interrupt-driven handshaking I/O, chapter 6 is the longest (52 pages) and most involved chapter. Since interrupt-driven systems are inherently complex, Goldsborough gives extensive attention to system timing considerations, interrupt service routines and associated hardware. Numerous illustrations explain the internal signal handling of mode 1 input and output operations. The book includes an example of a fully decoded keyboard and a video display, discusses the signal requirements and provides a program listing.

Concluding the example is a discussion of a design approach to using a PPI for mode 1 handshaking I/O. The discussion consists of guidelines that the reader should consider when designing a mode 1 PPI interface for any use, not just a keyboard and display. The five experiments are very comprehensive. As usual, schematics are given, in-depth questions are asked and the answers provided. Also, timing diagrams aid in comprehension.

Mode 2 (bidirectional I/O) operation is covered in chapter 7. The concept of master and slave CPUs is illustrated in this chapter. The discussion includes block diagrams, timing diagrams, operational sequence of events tables (those are invaluable to understanding the material), flowcharts and program examples. As always, the programs are well-documented to aid in understanding program flow.

The experiment illustrates the use of two 8080 microcomputers interfaced for bidirectional I/O, and shows the sequential transfer of data from the master to the slave and back. Unless you have access to another 8080-based system, merely reading the experiment must suffice. That is not to say, however, that nothing can be gained from the experiment if you don't happen to have an extra system. As with the other experiments, the reader can follow, and probably learn from, the experiment simply by reading the programs, circuit and timing diagrams and answering the questions.

Two appendices cover electrical characteristics and timing diagrams for the PPI and the 8255 control word and status word formats. I found that even with the PPI status and control word information included in appendix 2, it was convenient to dog-ear pages that explain the various format combinations. As with most other Sams books I've read, the index leaves something to be desired, although this one is better than most.

Although I thought I was familiar with the PPI operation, I learned quite a bit from the book. Even if you don't use an 8255 in your system, the material is presented in such a manner that you'll gain a familiarity with programmable peripheral interfaces. If you are designing an interface, you should be able to reduce the hardware complexity (and probably ease programming, too) after reading the book.

The book was well-written and well-composed. The examples and experiments not only reinforced the text, but also let the reader do some self-testing. To that end, the questions were for the most part in-depth enough to make the reader think further about the material.

J. C. Hassall
Blacksburg, VA

Program Conversions

Have you made any conversions lately? If so, *Kilobaud Microcomputing* wants to hear about them.

We're talking, of course, about converting programs found in the pages of *Kilobaud Microcomputing* for use on your system. Many fine programs have been published in the magazine, but they are of little use to you if they aren't compatible with your system.

If you have successfully converted a program for use on your system, why not send it in (listing and cassette or disk)? We'll check it out to see that it runs OK and prepare it for possible publication. This way, you'll be sharing your efforts with thousands of other users, who'll be able to add another selection to their program libraries. Of course, you will be rewarded handsomely for your efforts.

The home computer you thought was years away is here.

C8P DF

Ohio Scientific's top of the line personal computer, the C8P DF. This system incorporates the most advanced technology now available in standard configurations and add-on options. The C8P DF has full capabilities as a personal computer, a small business computer, a home monitoring security system and an advanced process controller.

Personal Computer Features
The C8P DF features ultra-fast program execution. The standard model is twice as fast as other personal computers such as the Apple II and PET. The computer system is available with a GT option which nearly doubles the speed again, making it comparable to high end mini-computer systems. High speed execution makes elaborate video animation possible as well as other I/O functions which until now, have not been possible. The C8P DF features Ohio Scientific's 32 x 64 character display with graphics and gaming elements for an effective resolution of 256 x 512 points and up to 16 colors. Other features for personal use include a programmable tone generator from 200 to 20KHz and an 8 bit companding digital to analog converter for music and voice output, 2-8 axis joystick interfaces, and 2-10 key pad interfaces. Hundreds of personal applications, games and educational software packages are currently available for use with the C8P DF.

Business Applications
The C8P DF utilizes full size 8″ floppy disks and is compatible with Ohio Scientific's advanced small business operating system, OS-65U and two types of information management systems, OS-MDMS and OS-DMS.
The computer system comes standard with a high-speed printer interface and a modem interface. It features a full 53-key ASCII keyboard as well as 2048 character display with upper and lower case for business and word processing applications.

Home Control
The C8P DF has the most advanced home monitoring and control capabilities ever offered in a computer system. It incorporates a real time clock and a unique FOREGROUND/BACKGROUND operating system which allows the computer to function with normal BASIC programs at the same time it is monitoring external devices. The C8P DF comes standard with an AC remote control interface which allows it to control a wide range of AC appliances and lights remotely without wiring and an interface for home security systems which monitors fire, intrusion, car theft, water levels and freezer temperature, all without messy wiring. In addition, the C8P DF can accept Ohio Scientific's Votrax voice I/O board and/or Ohio Scientific's new universal telephone interface (UTI). The telephone interface connects the computer to any touch-tone or rotary dial telephone line. The computer system is able to answer calls, initiate calls and communicate via touch-tone signals, voice output or 300 baud modem signals. It can accept and decode touch-tone signals, 300 baud modem signals and record incoming voice messages.

These features collectively give the C8P DF capabilities to monitor and control home functions with almost human-like capabilities.

Process Controller
The C8P DF incorporates a real time clock, FOREGROUND/BACKGROUND operation and 16 parallel I/O lines. Additionally a universal accessory BUS connector is accessible at the back of the computer to plug in additional 48 lines of parallel I/O and/or a complete analog signal I/O board with A/D and D/A and multiplexers.

Clearly, the C8P DF beats all existing small computers in conventional specifications plus it has capabilities far beyond any other computer system on the market today.

C8P DF is an 8-slot mainframe class computer with 32K static RAM, dual 8″ floppies, and several open slots for expansion.

Prices start at under $3,000.

Computers come with keyboards and floppies where specified. Other equipment shown is optional.

For literature and the name of your local dealer, CALL 1-800-321-6850 TOLL FREE.

OHIO SCIENTIFIC
a M/A-COM Company ✓4
1333 SOUTH CHILLICOTHE ROAD
AURORA, OH 44202 • (216) 831-5600